REMOTE DATA QUERYING

By
Bernie Ruane

OCS Consulting
REMOTE DATA QUERYING

■ AGENDA

– OCS COMPANY OVERVIEW
– PROJECT BACKGROUND
– CURRENT APPROACH
– PROPOSED SOLUTION
– RDQ DEMONSTRATION
– RDQ BENEFITS
– RDQ PROPOSITION
OCS STRUCTURE

OCS CONSULTING PLC - 1984

PROJECT MANAGEMENT & METHODS SERVICES

NATURAL SOLUTIONS 1986

DATABASE SOLUTIONS 1990

SAS SOLUTIONS 1995

SYPOWER SOLUTIONS 1996

BENELUX 1998

INSPECTION & TESTING SERVICES 1999

BUSINESS INTELLIGENCE & DATA WAREHOUSING SERVICES 1999

SUPPORT SERVICES 1999
OUR CURRENT BUSINESS
SAS SOLUTIONS

- ESTABLISHED 1995 - SPECIALIST IT SERVICES
- QUALITY PARTNER
  - UK in 1996
  - Benelux in 1999
- CURRENT SAS CONSULTANTS - 30
- BUSINESS PARTNERS: SAS, SEMA, SIEMENS
- LEADING UK SAS CONSULTANCY
- INTERNATIONAL FOCUS
REMOTE DATA QUERYING

- BACKGROUND
  - CLIENT REQUIREMENTS
  - APPROACHED BY SAS INSTITUTE
  - FORMAL APPROACH / METHODOLOGY
CURRENT APPROACH TO RESOLVING DATA QUERIES

- VALIDATION ROUTINES GENERATE QUERIES
- QUERIES WRITTEN ON A STANDARD FORM
- FORM SEND TO INVESTIGATOR BY FAX/POST
- INVESTIGATOR RESPONDS BY FAX/POST
REMOTE DATA QUERYING

- CURRENT APPROACH TO RESOLVING DATA QUERIES
  - SLOW AND EXPENSIVE
  - HUMAN ERROR ??
  - AVERAGE RESPONSE TIME TO QUERY 1 DAY - 1 WEEK
  - AVERAGE COST PER RESOLUTION/QUERY IS £100
  - RELY ON POSTAL SYSTEM/FAX
REMOTE DATA QUERYING

AIM TO BUILD A SOLUTION THAT:

- USES EXISTING VALIDATION SYSTEM
- SPEEDS UP RESOLUTION OF QUERIES
- IS COST EFFECTIVE
- IS EASY TO USE
- SUPPORTS ADDITIONAL FUNCTIONALITY
- USES LATEST WEB TECHNOLOGY
- ALLOWS REMOTE ACCESS
REMOTE DATA QUERYING

ADMINISTRATOR

E-MAIL
WEB BROWSER

plus...
Base SAS
SAS ACCESS TO...
SAS CONNECT

INVESTIGATOR

E-MAIL
WEB BROWSER
REMOTE DATA QUERYING

Remote Data Querying

Demonstration...
REMOTE DATA QUERYING

- USES EXISTING VALIDATION PROGRAMS
  - ADD ROUTINES TO CREATE
    - AN EMAIL TO THE INVESTIGATOR
    - THE APPROPRIATE QUERY
REMOTE DATA QUERYING

- SPEED UP QUERY RESOLUTION
  - NOT RELIANT ON ACCESS TO A FAX
  - NOT RELIANT TO RECEIVING POST
  - HOW QUICKLY EMAIL CAN BE ACCESSED?
REMOTE DATA QUERYING

- COST EFFECTIVE
  - IMMEDIATE RETURN ON INVESTMENT (ROI)
  - USE EXISTING SAS SOFTWARE
  - UTILISES EXISTING E-MAIL SYSTEM
REMOTE DATA QUERYING

- **EASE OF USE**
  - ELIMINATE LEARNING CURVE
  - SIMPLE USER INTERFACE
  - SPEED OF RESPONSE
REMOTE DATA QUERYING

- SUPPORTS ADDITIONAL FUNCTIONALITY
  - AUDIT TRAIL
  - AD-HOC REPORTING
  - PERFORMANCE ANALYSIS
  - DATA MINING
REMOTE DATA QUERYING

- SECURITY
  - MINIMAL DATA BEING ACCESSED VIA E-MAIL
    - PATIENT NUMBER
    - TRIAL I.D.
    - QUERY ERROR
  - UTILISE EXISTING FIREWALL
  - ALLOWS FOR ADDITIONAL SECURITY
REMOTE DATA QUERYING

- IMMEDIATE RESPONSE
- MINIMAL COST PER QUERY
- USABILITY
- TRANSITION
- IMPLEMENTATION
- WEB/E-MAIL
- LOW COST
REMOTE DATA QUERYING

Remote Data Querying

PROPOSITION
REMOTE DATA QUERYING

Customise RDQ
- identify
- data feeds
- functionality
- security

Install Core RDQ

Scoping Requirements Document

Support

Implementation

User Testing

Testing
- System
- Integration

Implement in U A T Environment

Proof of Concept

Customisation of RDQ
- identify
- data feeds
- functionality
- security

Support

Implementation

User Testing

Testing
- System
- Integration

Implement in U A T Environment
QUESTIONS
Efficiency Using Enterprise Reporter ®
Nikkie Windsor - OCS Consulting

Agenda

The following points will be covered during this presentation:

- OCS Overview
- Client Requirements
- Why we chose Enterprise Reporter ®
- Development Techniques Used
- Efficiency Lessons Learnt
- Questions and Answers

Client Requirements

Our client required a combination of many different features to be present in the solution provided by the chosen reporting tool. The solution was also to be implemented under an extremely aggressive time scale, as the requirement was for a reporting tool to be in place in a number of weeks.

Standard layout reports were required, to provide a number of default reports to every user, with minimal effort involved in obtaining the current business status. Within these reports, the flexibility must also be available to alter layout and content, and filter data as required. Ad hoc queries were required against the existing database structures. These queries needed to be quick and easy to generate.

Our client wanted the reporting tool to be intuitive to use, for all levels of user, including those without in-depth technical knowledge. Users were to be able to easily filter the data displayed in a report. It should also be possible to generate complex filter criteria to identify the subset of data required. There were to be a variety of different presentation styles available to a user, including a number of different graphs and charts, and multidimensional tables to depict data hierarchy. There was a requirement to be able to generate HTML documents from reports. These HTML documents should be cosmetically pleasing, with straight forward navigation.

The overall requirement was for the end result to have the power and versatility of PROC TABULATE and PROC GCHART, with the flexibility and ease of use of a Microsoft Office application.

The existing IT Infrastructure involved an existing Client Server architecture. Each client was running Windows 95, and Servers running UNIX and Windows NT were used. A number of different database environments were also used, including SAS and Oracle. The processing power of the Servers involved was to be utilised to full advantage. Also, access to the data on different platforms was to be transparent to the user of the reporting tool. Access to all data should be uniform in approach and presentation. The reporting tool was required to integrate into existing SAS/AF and SAS/EIS applications.

Within the reporting tool, restricted data access on a User by User basis, and a User Group basis was required. Altering user access needed to be straight forward, to minimise the level of administration required. It also needed to be straight forward to publish new standard reports into the production environment, to be accessed by authorised users.
Why we Chose Enterprise Reporter ®

A number of reasons influenced the selection of Enterprise Reporter ® as the chosen solution for our client. It is possible to configure and install Enterprise Reporter ® software, and User and Group definitions quickly. Report generation is also extremely rapid.

Within Enterprise Reporter ® it is possible to publish a set of standard reports, which could be selected from a set of icons in a Report Gallery environment. These reports could be attached to data, or in a template format, and are freely modifiable. It is also straight forward to perform ad hoc queries against data, and to save those ad hoc queries as reports. Enterprise Reporter ® Filtering interface enables fast definition/alteration of filters, with data driven selections. It is also possible to build complex Where Clause subset criteria using this interface.

Enterprise Reporter ® is specifically designed as an end user reporting tool. The Microsoft Office Look and Feel environment provides a familiar and intuitive interface for most users. A selection of report icons from a Gallery environment simplifies the interaction for users with little technical and/or business data knowledge. New reports are built using Drag and Drop techniques, which also simplifies user interaction. Also, a Print Preview mode is provided, giving an accurate representation of printed output.

Enterprise Reporter ® provides a variety of graphical and tabular structures for building reports. Combinations of these structures can be used in the same report, to obtain multiple-layout reports. Structures are configurable, and all standard features, such as colours and fonts, are changeable. Within Enterprise Reporter ® it is possible to wrap a report with standard document sections, such as headers, footers, title pages and summary pages, to create a complete document.

HTML documents can be easily created from Enterprise Reporter ® reports, by using the Save As option, as with Microsoft Office applications. HTML documents layout is structured in such a way as to make viewing easy from a Browser. The navigation through the documents is aided by creation of Hypertext Links, in the form of a table of contents depicting data categories within the report.

Enterprise Reporter ® fully exploits the Client Server architecture, allowing time consuming queries to be performed by the server, with only the result of the query being passed back to the client. Connection to servers is dealt with automatically by Enterprise Reporter, and the platform and database environment from which data are drawn can be transparent to the user. Data from any combination of platforms and database environments can be used to drive a single report. Enterprise Reporter can automatically join data by common key fields, in order to simplify use of normalised database structures (as in Relational Databases). It is also possible to explicitly define the join criteria between tables.

Enterprise Reporter ® can be invoked from SAS/AF and SAS/EIS applications, with a simple AFA command, enabling easy integration into existing SAS built business applications.

Enterprise Reporter ® provides a User and User Group definition mechanism, whereby user access can be restricted to both data and reports. Altering user access is straight forward, by altering user group membership. New reports can be published by simply placing the report files in the correct directory for inclusion in the Report Gallery.

The latest version of Enterprise Reporter ® (Release 2.0 Service Pack 1) was used, as a number of new/enhanced features were relevant to our client’s requirements. Oracle access has been enhanced. Access is now possible without providing an Oracle path for the required instance. Also, Oracle views that are available to the user are included in the available report data. This release of Enterprise Reporter ® includes two new graph types: Spline and Join. Enhancements have also been made to the HTML Output of large list or table objects.
**Development Techniques Used**

There are a number of built in methods for enhancing efficiency within the Enterprise Reporter ® environment. The following features/restrictions were used to ensure maximum performance from the developed reports:

**Restricting Report Data**

From the available data within the Info Center, selections of columns was restricted to just those columns that would actually be used within the report. This ensured that the number of data items obtained by Enterprise Reporter ® generated queries was kept to a minimum. The filter window was used to subset the data retrieved from any query.

**Query Governor**

The query governor was used to place restrictions on queries that users may perform, thus prohibiting the generation of large queries that may affect the performance of the IT environment. The number of Input Rows to, and Output Rows from a query were both restricted. Processing cycles used by a query were also restricted, to ensure that the server response time remained within acceptable boundaries.

**Report Optimiser**

Report Optimiser was used to restrict both the number of data items in a report, and the sample size of data used when displaying a report. With these options set, any reports containing a large number of group/category variables, or requiring a large sample of data, produced warning messages.

**Totals and calculated columns**

Enterprise Reporter was used to generate Totals and Sub-Totals within table objects. Calculated columns were generated from existing items in the report data, where required data were not stored in the existing data, but could be derived. Both totals and calculated columns were found to have little impact on performance when generating reports.

**START.SAS and END.SAS**

These programs were added to the directories that held InfoFolder definitions, and were used to carry out a number of different tasks. User-defined formats that were used for data within reports were generated. Macro variables that were used within the Enterprise Reporter ® environment were defined. Also, large data tables had subsets applied prior to use in reports.

**Invocation of Enterprise Reporter ®**

Enterprise Reporter was invoked in a number of different ways, dependent upon the users the system was delivered to, and their access to existing SAS Developed applications. Where users had access to an existing custom developed AF application, access to Enterprise Reporter ® was provided through the existing application interface. The following command structure was used to invoke Enterprise Reporter ® from SAS, in the same way an AF application is invoked:

```
AFA C=sashelp.business.start.scl
   COMMAND='GALLERY'  (Open Report Gallery)
   USER='<User ID>'
   EXITICON='NO'     (Do not minimise when existing)
```

Where users had access to an existing SAS/EIS Application, access to Enterprise Reporter was provided through the existing EIS application interface. The command structure to invoke Enterprise Reporter ® from an EIS
application is the same as the above command. It is necessary to ensure that the SAS configuration file for the EIS application sets the BUSROOT library, and includes the SASEXE Directory below this location in both the SASHELP Library definition, and the image search path (PATH) definitions, as in the following example.

Excerpt from CONFIG.SAS File:

```
-SET BUSROOT "C:\SAS\BUSINESS"

/* Setup the SAS System help directory definition */
-SASHELP
  (  
    !SASFOLDER\SASCFG
    !sasext0\af\sashelp
    !sasext0\assist\sashelp
    .
    .
    !sasext0\stat\sashelp
    !sasext0\mddbserv\sashelp
    !BUSROOT\SASHHELP
  )

/* Setup the SAS System load image search paths definition */
-PATH
  (  
    !sasext0\af\sasexe
    !sasext0\assist\sasexe
    !sasext0\base\sasexe
    .
    .
    .
    !sasext0\stat\sasexe
    !sasext0\mddbserv\sasexe
    !BUSROOT\SASEXE
  )
```

Users without access to an existing AF EIS application, where given access directly to Enterprise Reporter®, via an icon on the PC desktop.
Efficiency and Technical Lessons Learnt

It was found that there were some possible areas for efficiency/performance tuning, and report tailoring, within the reporting application we developed for our client. The following key areas were addressed during our development projects:

Client Server Architecture

Where data were accessible both from the Client and Server (for example, on an NT Server, where network mapped drives provide client end access to data on the server), we found a significant improvement in performance by opening a Client Server connection, and allowing the server to process large volumes of data. If data volumes were small, then performance would be aided by allowing Enterprise Reporter to access the data locally, removing the additional time required for Client Server communications.

Pre-joining Data

Although Enterprise Reporter ® is capable of automatically joining disparate data sources, we found that it is sometimes more efficient to pre-join these data prior to processing. This can be done by creating views or intermediate summary tables, incorporating the required elements from the data. Views can be created in the Server database environment, and are then queried by Enterprise Reporter ® in an identical manner to tables.

Enterprise Reporter Batch Mode

Some users merely required a daily updating static image of the standard reports provided. For these users, it was sufficient to provide a non-interactive document, consisting of the default contents of the standard reports. Where this was the only requirement, it was felt that there was unnecessary delay involved in opening Enterprise Reporter ®, when little of the functionality would be used. The delivery of instant-response reports for these users was achieved by using Enterprise Reporter ® in batch mode, to generate HTML documents from the Enterprise Reporter ® reports. These documents were then accessed via shortcuts on each user’s PC desktop. The following command structure was used to generate HTML reports in batch mode:

C:\EntRep\Business\ERSAS.EXE
  HTML="<Full Report Path>"
  USER=username
  BATCH=yes
  SASPATH=<SAS Executable Directory Path>
  OUT=<Full HTML Output Document Path>
  SASPARM="-ALTLOG <SAS Log Full File Path>"

Subset Metadata using ORARead.SAS

The ORAread.SAS program contains code that retrieves metadata from the Oracle dictionary tables. A customised ORAread.SAS file is supplied with Enterprise Reporter ® and can be customised further as required. To utilise this program file it must be defined when creating the InfoFolder. If it is not used, all oracle tables and views in the current Oracle subsystem that you have read access to are included in the metadata. Basic sub-setting can be achieved by changing the value of macro variable WHECRE. This is used in the where clause against the OWNER column in DICTIONARY views. Metadata will be created for the tables owned by the specific Oracle user defined by OWNER.

More advanced sub-setting can be written into the SQL queries that generates the metadata but this level of customisation will require some knowledge of Oracle and SQL.

The user will only see Oracle tables that they are allowed to access.
If no Oracle primary keys or foreign keys, the relations must be defined manually.

**Using Prompt for Oracle Date filtering**

Anything specified in the Filter window in Enterprise Reporter ® is used to build a Where clause in the SQL query generating the report. This filter window only accepts criteria recognised by SAS and will therefore only recognise SAS date functions.

Any Where clause that is passed to Oracle that contains SAS functions will not be recognised. The Where clause will therefore be passed to SAS for execution.

By using Prompt, you can assign an Oracle Date function value to a macro variable in the ‘Define Prompt’ option on the Tools menu. Enter into the macro the value you wish to filter by in the filter window. This macro is resolved in the Oracle query and is recognised by the DBMS. The Where clause is executed in Oracle.

Do not use Prompt for Oracle date filtering if there are any SAS functions, SAS variables or calculation variables being used in the filter. These SAS values or variables will not be recognised by Oracle and the Where clause will be executed by the SAS system. If you are using an Oracle date function in Prompt when there is other SAS filtering, the Oracle syntax will not be recognised when the Where clause is executed by SAS and processing will stop with an error.

**Using SAS/MDDB**

If Enterprise Reporter ® uses SAS/MDDB, it tends to generate its own MDDB as it processes. We found that it is much more efficient to use a pre-generated MDDB to drive reports within Enterprise Reporter.

**Creating calculations on Detailed or Summarised Data**

It is possible, when defining calculated columns, to specify whether the calculation is carried out at the detailed data level, or once the data have been summarised to the level that the report displays. One calculation gives a simple average calculated on detailed data whilst the other gives a weighted average calculated on summarised data. The results can be different.

**Conditional Formatting and Traffic Lighting Data**

It is possible to conditionally format data within table objects, specifying different values for the data attributes, dependent upon ranges of data values. It is possible to conditionally apply such formatting as font size, type and name; text colour; and background colour. This enables effective traffic lighting of data within a table.

**Creating additional formulae for Calculations**

There is a default list of formulae available to the user when creating calculations in Enterprise Reporter ®. It is possible to add to this list of formulas with user defined formulae:

- create a saved SCL list containing your formula skeletons
- store the list in a catalog called BUSCALC and have the entry name CALCULAT.SLIST
- concatenate the folder with this catalog to the SASHELP search path in the CONFIG.SAS file

Enterprise Reporter ® automatically appends the formulas in the list with the formulas supplied with the software.