

Paper 018-2011**Extending the Reach of SAS® Metadata**
Edward Lambe, Central Statistics Office, Cork, Ireland**ABSTRACT**

Most companies struggle to provide an end-to-end metadata solution for the data assets they hold. The descriptor portion of a SAS® data set can be an important starting point in delivering this solution. By leveraging this data, we can start to provide a complete picture of our data assets.

This paper demonstrates the steps required to start building that end-to-end metadata solution, highlighting the use of PHP, Base SAS®, SAS ODBC, and FLEX to build a prototype solution.

INTRODUCTION

Most companies struggle to provide an end-to-end metadata solution for the data assets they hold. The descriptor portion of a SAS Data set can be an important starting point in delivering this solution, and by leveraging on this data we can start to provide a complete picture of a company's data assets. This paper looks to demonstrate how to develop a web based solution that will build on the power of the SAS data dictionary tables to create a central metadata store.

For this presentation I have assumed that a company needs to capture information under the following headings: (These can be extended to meet your business requirements.)

DATA SUMMARY

- Containing SAS and Company defined metadata
- Data Set Metadata
 - Create / Modify Date, Number of Observations, Number of Variables and Data set Size
- Company Metadata
 - Data custodian, Telephone, Email and Short Summary
- Data Set Column Metadata
 - Column Name, Label, Type, Length, Informat, Format

DETAILED DESCRIPTION

- A HTML Compliant Data Editor where users can enter detailed information regarding their data sets

SUPPORTING DOCUMENTATION

- Users can associate documentation against their data set providing a summary as to why the information is deemed relevant.

USER ACCESS

- User access to the data sets and the purpose for that access can be recorded for each data set or library.

TECHNICAL OVERVIEW

The solution employs the following technology elements: Adobe Flex, PHP, Apache, SAS ODBC and Base SAS. The solution developed utilizes a front-controller design pattern. All requests are routed to the PHP front controller and in turn these are dispatched to query the SAS dictionary tables via the relevant controller. All data retrieved is parsed into JSON format and returned to the end-user. An overview of the process flow is as per the following diagram:

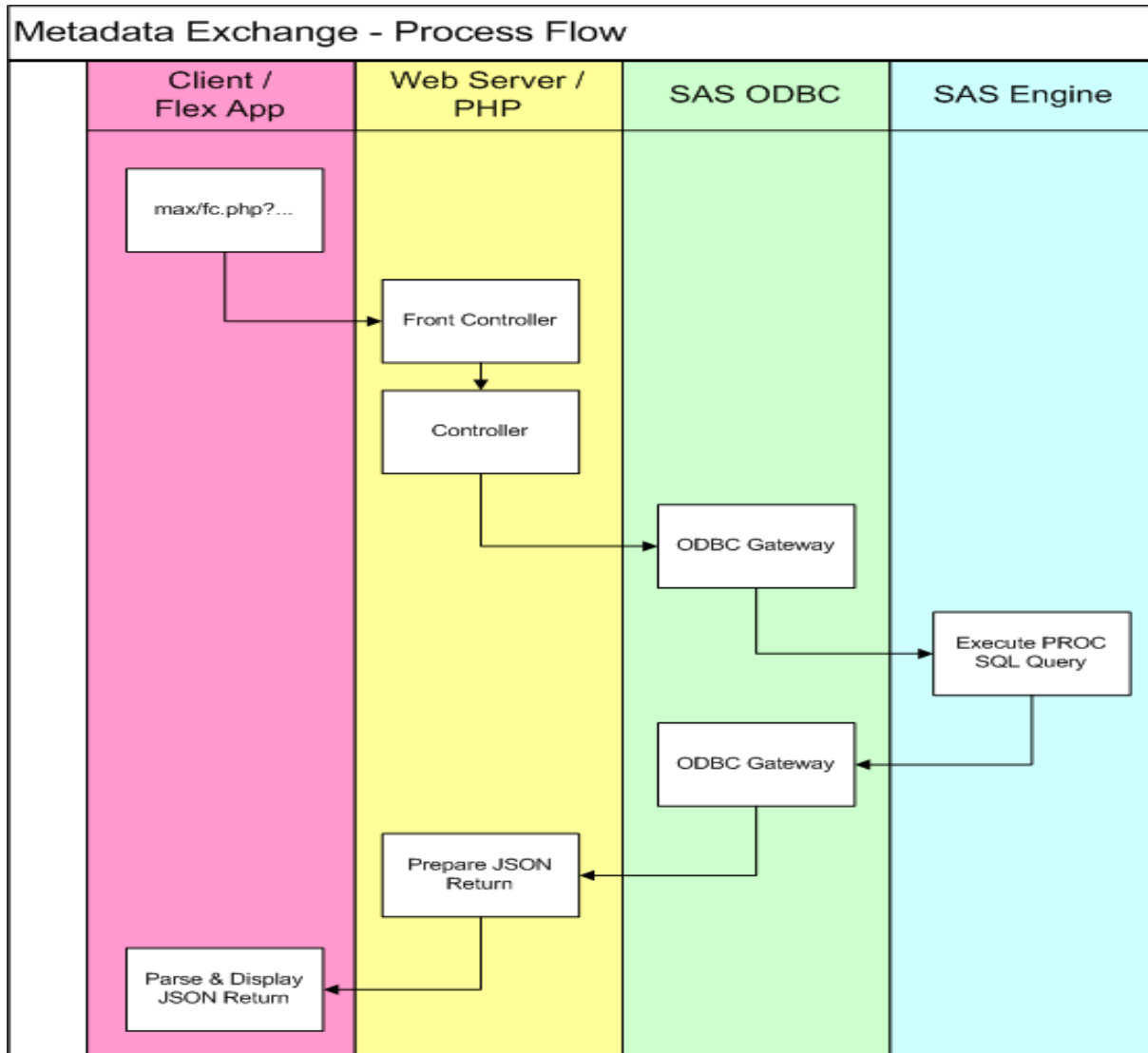


Figure 1: Process Flow

THE SAS COMPONENTS: SAS ODBC & BASE SAS

Access to the SAS data sets is gained via the SAS ODBC Driver. The SAS ODBC driver provides a gateway through which data requests are forwarded to the SAS application engine for processing.

The SAS ODBC driver is configured as follows:

1. Install the SAS ODBC Driver
2. Create a SAS ODBC Data Source (Admin Console / Data Sources)

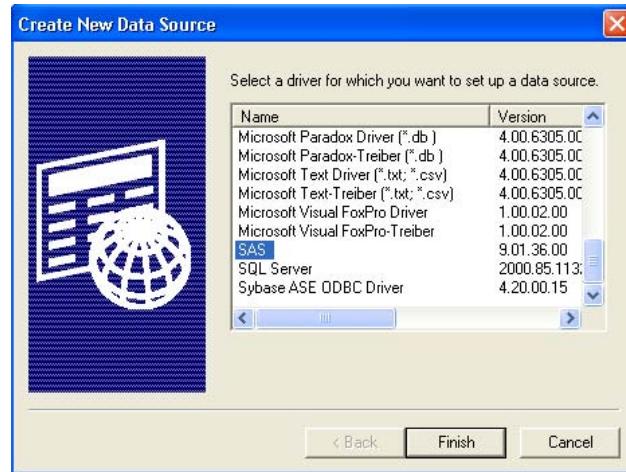


Figure 2: Defining a SAS ODBC Data Source

a. Step 1: Define a Server

Please refer to the SAS ODBC Driver documentation on the SAS website for detailed information on this section. Note depending on your configuration, i.e. local or remote server you may need additional SAS software and licenses.

b. Step 2: Define a Data Source

Provide a Data Source Name, Descriptions and the Server to which the ODBC driver will connect.

c. Step 3: Define SAS Libraries (Figure 3)

In the Library Settings section define the Name (This is the name that will be assigned to your library

d. Complete the Host File field – this is the directory in which the SAS data sets reside.

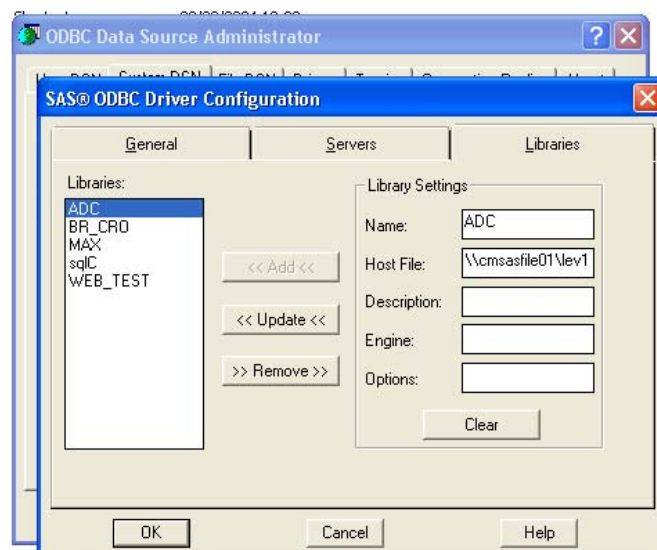


Figure 3: Define SAS Libraries

3. Start the SAS ODBC service

Open a SAS session and execute the SASODBC macro passing the server name you wish to start. Detailed information regarding the configuration options is available on the SAS website and in the SAS ODBC Manual.

WEB SERVER

The web server solution runs on a Windows / Apache / PHP platform. For the prototype developed a central application object was created which reads the relevant configuration files, creates the required database connections and initiates the logging mechanism for the solution. This application object is then extended by our front controller object.

Client data requests are forwarded to the front controller and the data request is routed to the relevant controller / helper. The PHP code interrogates the SAS data dictionaries using SQL queries and returns a JSON (Java Script Object Notation) string to the client for display.

There are a number of options available with PHP on interrogating an ODBC data source. In the development of the prototype the PEAR:DB library was utilized.

CLIENT INTERFACE

The prototype client interface was created using the Adobe Flex Rich Internet Application (RIA). The Flex language allows developers to create an engaging user interface that can be deployed through the browser using Adobe Flash, or through the desktop using Adobe AIR.

In the prototype developed the client interface is used to display and return user input to the PHP web server for processing. The following section provides detail as to the working of the prototype solution.

DEMO OF THE METADATA EXCHANGE APPLICATION

The application contains two functional sections (Figure 4); the Application Control Bar is the primary navigation mechanism, and the Application Body where users view and enter metadata relating to their selected data set.

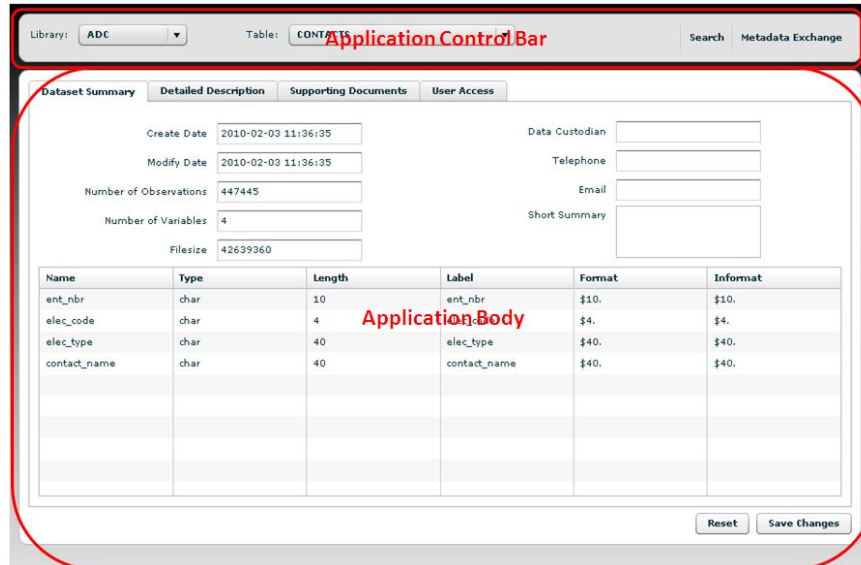


Figure 4: Application Layout

On loading the application loads all the libraries that have been defined by the SAS ODBC Data Source (Figure 5). (Here we are querying the 'dictionary.libnames' table.)

By default it displays the tables for the first library in the collection at startup.

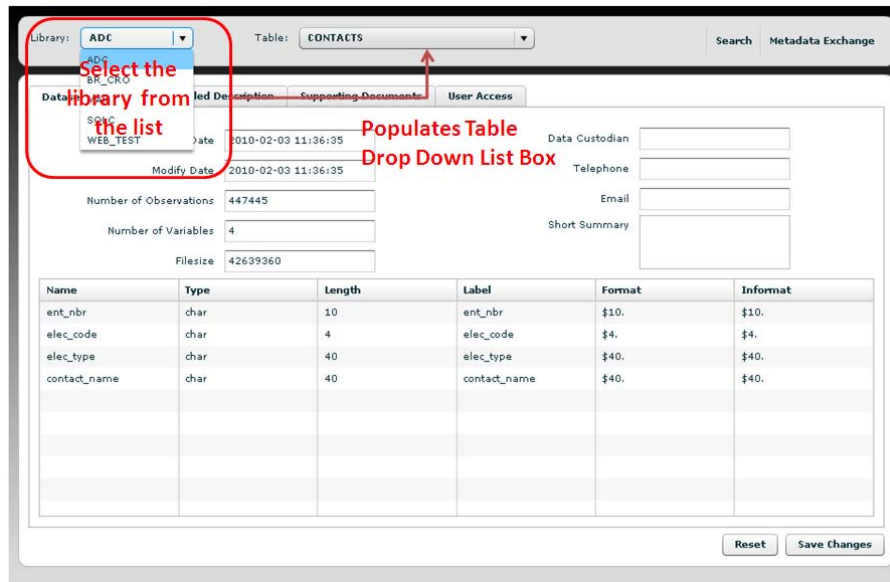


Figure 5: User selects library from list

The user can then select the SAS data set they wish to view (Figure 6) which returns the relevant metadata associated with the table to the application body. (Here we are querying the 'dictionary.tables' and 'dictionary.columns' tables.)

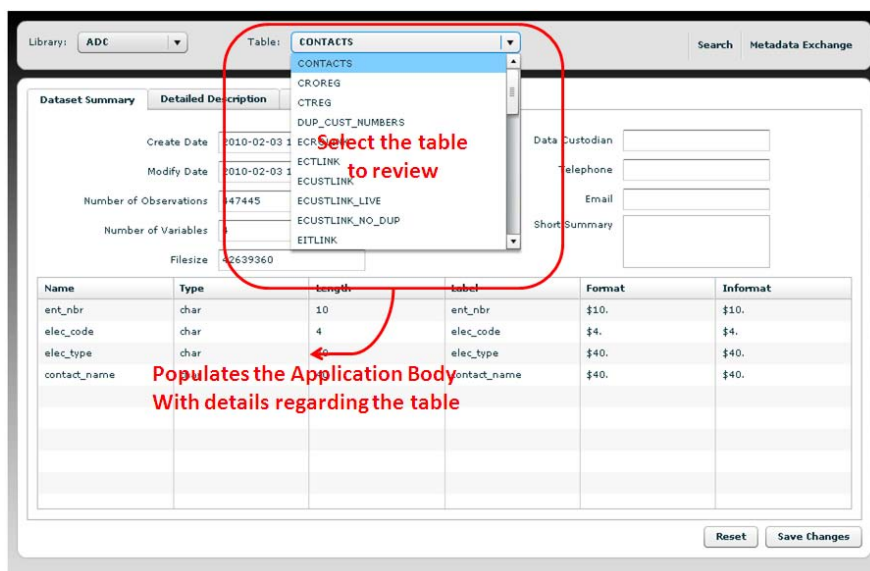


Figure 6: User selects data set to review

The Data set Summary (Figure 7) screen contains three areas of interest to the user:

1. The SAS Metadata

- This contains metadata relating to the table and displays the creation & modification date, number of observations, number of variables and the file size on disk for the data set selected.

2. CSO Metadata

- This allows us define who 'owns' the data set. We can enter the data custodian name, telephone, email and a short summary of what is contained within the data set.

3. SAS Column Metadata

- This section displays the column metadata associated with the SAS data set, displaying column / variable name, type, length, label, format and informat.

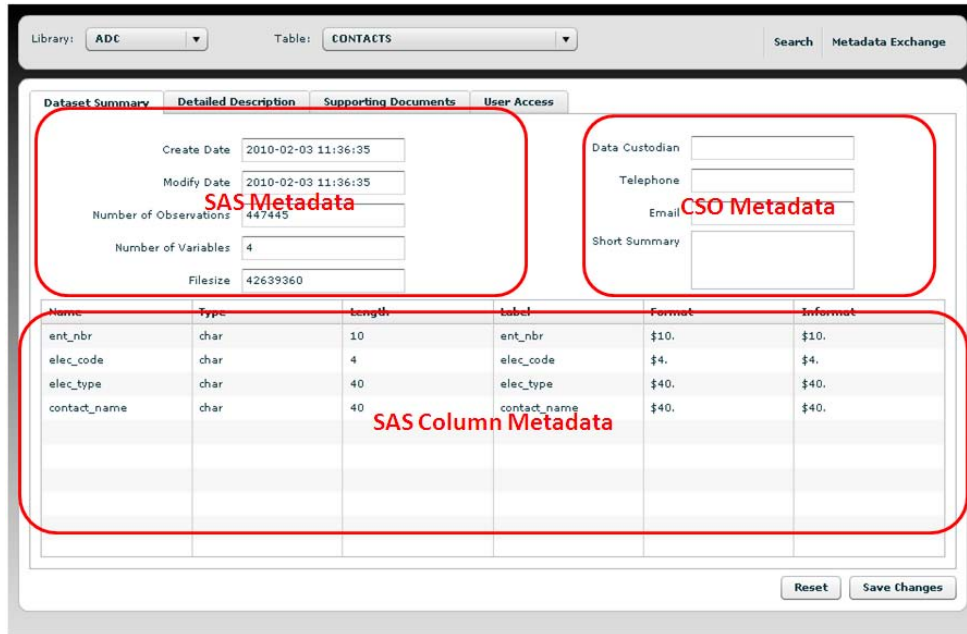


Figure 7: Data Summary Screen (SAS & Company metadata returned)

The Detailed Description section (Figure 8) provides a HTML Compliant Data Editor which allows users enter a detailed description for the data set.

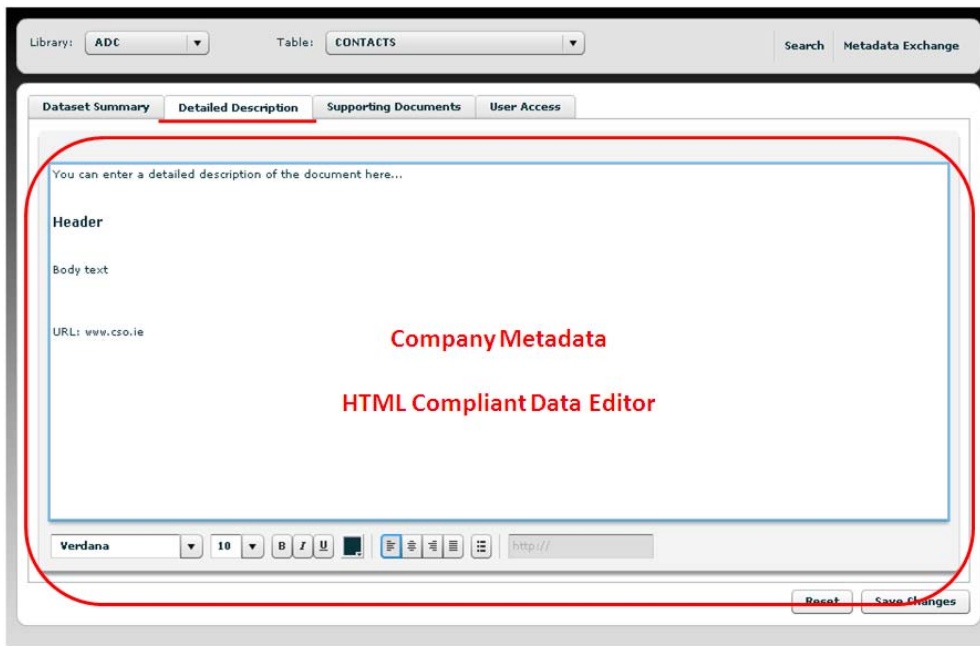


Figure 8: Detailed Description

The Supporting Document section (Figure 9) allows users to upload relevant element which should be associated with the data set. This can include system documentation, raw data used to populate the data set, relevant legislation or regulations pertaining to the data set, etc.

Users are asked to provide a short summary of why it is relevant to upload the file – this allows other users to quickly identify their information need at a later date.

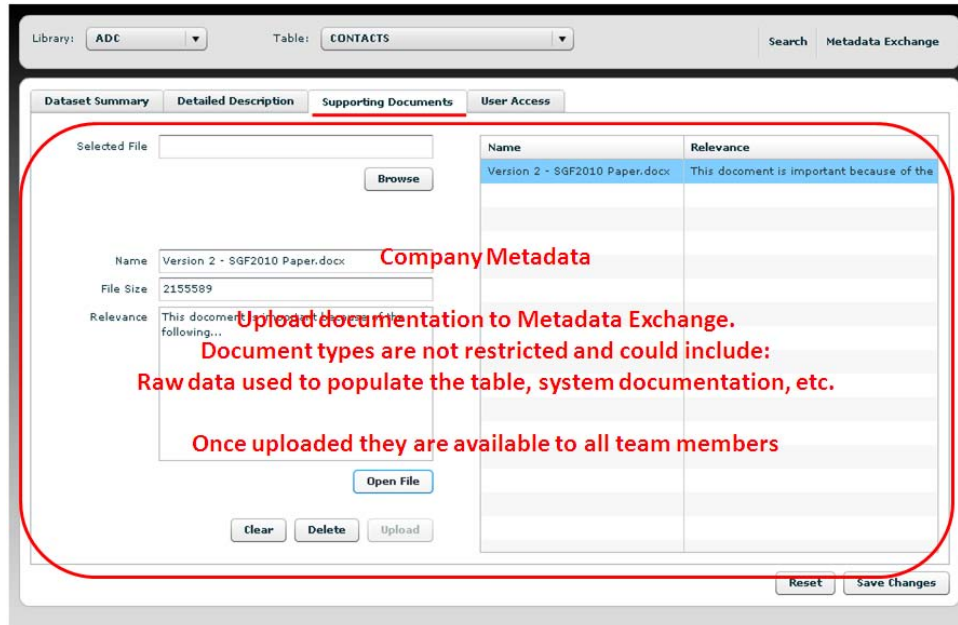


Figure 9: Supporting documentation saved against data set

The User Access section (Figure 10) allows the data custodian to define who should have access to the data. This can be linked to the network level security to ensure that data assets are securely managed.

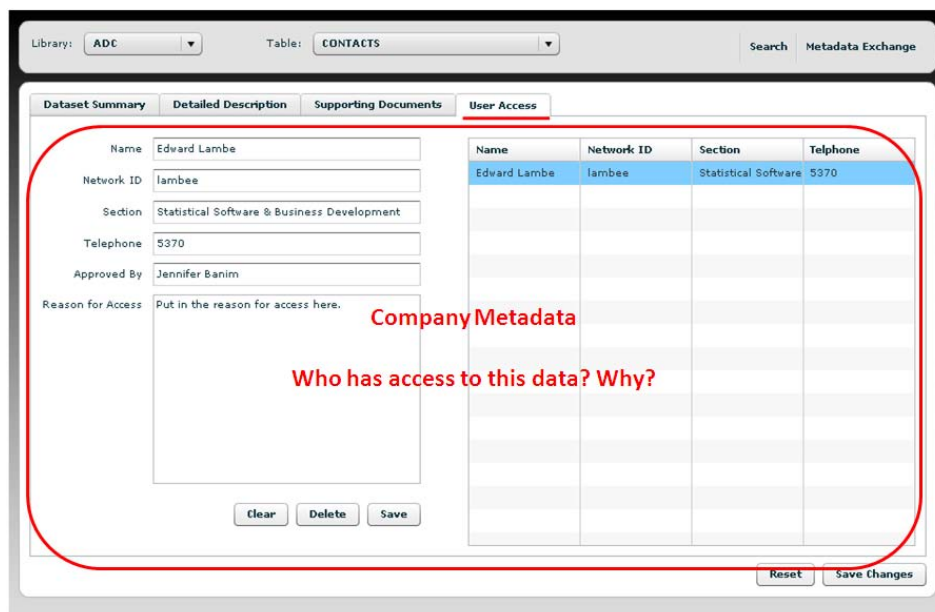


Figure 10: User Access to the data set

The search function (Figure 11) is useful function for users as they can quickly identify data assets that may contain the information they require.

The search function searches against all libraries and tables that are defined within the application and search the label and variable name for the search term entered.

Search Term:

Library	Data Set	Variable Name	Label	Type	Length	Format	Informat
MAX	TABLES	nobs	Number of Physical Observations	num	8		
MAX	TABLES	obslen	Observation Length	num	8		
MAX	TABLES	delobs	Number of Deleted Observations	num	8		
MAX	TABLES	nlobs	Number of Logical Observations	num	8		
ADC	EVATLINK	vat_nbr	vat_nbr	char	10	\$10.	\$10.
ADC	VATREG	vat_nbr	vat_nbr	char	10	\$9.	\$9.
ADC	VATREG	vat_cesdate	vat_cesdate	num	8	DATETIME22.3	DATETIME22.3
ADC	VATREG	vat_name	vat_name	char	70	\$70.	\$70.
ADC	VATREG	vat_title	vat_title	char	10	\$10.	\$10.
ADC	VATREG	vat_suffix	vat_suffix	char	10	\$10.	\$10.
ADC	VATREG	vat_com_name	vat_com_name	char	20	\$20.	\$20.
ADC	VATREG	vat_addr1	vat_addr1	char	35	\$35.	\$35.
ADC	VATREG	vat_addr2	vat_addr2	char	35	\$35.	\$35.
ADC	VATREG	vat_addr3	vat_addr3	char	35	\$35.	\$35.
ADC	VATREG	vat_addr4	vat_addr4	char	35	\$35.	\$35.

Searches all Libraries and Tables within the Metadata Exchange for your search term

Figure 11: Search function queries all data sets

CONCLUSION

The solution outlined is a prototype developed to show how easy it is to leverage on the SAS data dictionary to build a central metadata store.

The prototype also highlights the ease of integrating SAS with PHP, and shows how companies can quickly integrate the power of PHP and SAS to allow companies to build web-based applications for end-user interaction with SAS data set.

The prototype can be extended and developed further to create a data capture / editing function for SAS data sets. As users move away from PC SAS to the SAS BI Platform the FSEDIT procedure is no longer available; the functionality provided by the FSEDIT procedure can be replicated quickly via a PHP / SAS ODBC application.

REFERENCES

SAS ODBC Documentation

(<http://support.sas.com/documentation/onlinedoc/odbc/index.html>)

SAS Dictionary Tables

(<http://support.sas.com/documentation/cdl/en/lrcon/62955/HTML/default/viewer.htm#a002299818.htm>)

PHP

(www.php.net)

PEAR::DB

(<http://pear.php.net/package/DB/>)

Adobe Flex

(<http://www.adobe.com/products/flex/>)

CONTACT INFORMATION

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

Name: Edward Lambe
Enterprise: Central Statistics Office
Address: Skehard Road
City, State ZIP: Cork, Ireland
Work Phone: +353 453 5370
Fax: +353 21 453 5555
E-mail: edward.lambe@cso.ie
Web: www.cso.ie

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.