Paper BB-02

Loading Metadata to the IRS Compliance Data Warehouse (CDW) Website: From Spreadsheet to Database Using SAS® Macros and PROC SQL

Robin Rappaport, IRS Office of Research, Washington, DC
Jeff Butler, IRS Office of Research, Washington, DC

ABSTRACT

As the largest database in the Internal Revenue Service (IRS), the Compliance Data Warehouse (CDW) provides data, metadata, tools, and computing services to hundreds of research analysts whose work is aimed at improving tax administration. CDW manages an online repository of metadata to help users quickly search for and understand the meaning of data. With more than 25,000 unique columns and over 500,000 separate attributes, these metadata represent a key component of the overall user experience, and are part of a broader data quality initiative. In addition to standardized column definitions that are searchable through the CDW website, metadata also include lookup reference tables, data types, legacy source information, and other attributes. Most metadata are initially created or updated in Excel, after which they are imported into SAS data sets for additional processing. SAS macros are implemented to iteratively read, process, format, and perform other operations one column at a time for each table being updated. Using the ODBC engine type via the LIBNAME statement, PROC SQL is used to execute INSERT, UPDATE, or DELETE statements in Microsoft SQL Server, which stores the final metadata that is published to the CDW website. SAS logs are monitored to ensure the integrity of database transactions, and a staging website is analyzed to validate results. By leveraging the power of SAS to import source data, iteratively process data with macros, and update external databases, hundreds of IRS researchers can reliably access metadata on a regular basis to support their analytical needs.

INTRODUCTION

This paper explains how the IRS uses SAS to load metadata to the Compliance Data Warehouse (CDW) website. The CDW website gives researchers the ability to quickly search for and understand the meaning of data available for analysis. The power of SAS is leveraged to import data from Microsoft Excel spreadsheets to SAS datasets, iteratively process with SAS macros, and insert columns and other attributes into a Microsoft SQL Server relational database using PROC SQL and ODBC via a LIBNAME statement.

Compliance Data Warehouse (CDW)

The Compliance Data Warehouse (CDW) provides data, tools, and computing services to the IRS Research community and maintains a high-performance analytical environment for conducting research studies. It is the preferred platform for IRS research analysts and other business users to perform a variety of analytics on massively large amounts of tax data.

CDW is used by IRS research analysts for both short- and long-term studies. Examples include studying trends in defaulted installment agreements, offers in compromise, liens, and filing delinquencies; analyzing patterns in electronic payments, refundable credits, return preparers, and math errors; and tracking distributional characteristics of revenue collected from tax assessments and penalties. It is also used for estimating factors contributing to the U.S. tax gap; measuring the relationship between customer service and voluntary compliance; simulating the effects of policy changes on tax behavior; developing models for identifying compliance risk; producing estimates of taxpayer burden; and identifying abusive tax shelters and other schemes.

Due in part to the amount of online historical data it maintains, CDW's value has extended beyond the Research community to support over a dozen IRS modernization initiatives and enterprise strategies. It has been host to multiple prototype or proof-of-concept efforts, including the development of new employment tax workload allocation processes; a modernized Identity Theft reporting application; a new Correspondence Management Information System; and a new analysis and reporting system for taxpayer outreach. It has also supported multiple enterprise strategies for metadata and business intelligence.

CDW DATA QUALITY INITIATIVE

IRS source data that is captured and integrated in CDW are often subject to inaccuracies, anomalies, and other imperfections that are not always identified and treated at the source. As a result, IRS research analysts using data in CDW must be on the constant lookout for any data quality problems that might affect their results.

Although CDW does not have a dedicated staff of data analysts whose sole responsibility is data validation (and developing or updating validation rules), a model of stewardship within the IRS Research community has provided an effective alternative. Through their everyday use of CDW data, IRS researchers sometimes identify data values out of range, excessive missing values, overflow conditions, and inconsistencies in data typing or column naming conventions. Most research analysts take time to report problems to CDW staff, who then share them with the Research community via the CDW website or other communication channels and take corrective actions, where appropriate.

In general, CDW attempts to operationally address data quality in six areas:

1) Timeliness -- Minimize amount of time to capture, move, and release data to users; leverage new technologies or processes to increase efficiencies throughout the data supply chain.

2) Relevance -- Ensure data gaps are filled; make new investments in data that reflect expected future priorities.

3) Accuracy -- Create processes to routinely assess fitness of data; report quality assessment findings; publish statistical metadata through the CDW website; cross-validate release data against source and other system data.

4) Accessibility -- Improve organization and delivery of metadata; provide online knowledge base; facilitate more efficient searching; make new investments in third-party tools to simplify process of accessing and analyzing data.

5) Interpretability -- Standardize naming and typing conventions; create clear, concise, easy to understand data definitions.

6) Coherence -- Develop common key structures to improve record matching across database tables; ensure key fields have common names and data types.

CDW WEBSITE AND DATA PROFILING

The CDW intranet website provides metadata, data schemas, profiling capabilities, summary reports, data alerts, and other information that is useful to IRS research analysts. Metadata can be accessed by drilling down on successive links through a database-table-column hierarchy, or through free-form search.

In 2009, a radical new approach to data profiling and data discovery expanded metadata and search capabilities by including the ability to generate frequency tables, statistical distributions, trends, and geographic maps for virtually any column in the database—all executed in real time against hundreds of millions and even billions of rows of data. Basic filters, grouping, and drill-down options exist for each analytic function, making the number and combination of ad-hoc queries almost limitless. This innovative, custom-developed capability—the only one of its kind in the IRS—generates nearly 20,000 ad-hoc queries per month and has quickly grown to represent over 30% of all CDW database queries.

CDW METADATA

Metadata are critical to understanding the meaning of data, and are the foundation for the interpretability dimension of data quality outlined above. They are an important part of any process that turns data into usable information. They can be used as a tool to select appropriate data for analytics and understand and interpret findings. Metadata can also be used as a basis for developing business rules to validate and augment data.

CDW maintains a web-based repository of metadata for over 25,000 columns of data and more than 500,000 separate attributes. Metadata are available at the database, table, and column level, and are created and updated using internal reference material. Database and table-level metadata include a name, description, source, availability, update status, and links to other internal web sites for program or operational information. Column-level metadata include a definition, legacy reference, availability, release frequency, data type, primary key candidate, nulls, distribution type, range type, and other attributes. Metadata also include lookup reference tables displaying valid values and meanings for coded fields.

A standard template is used to create column-level metadata. It contains reference to the legacy source. It is informative, clear, concise, and complete. It is semantically consistent and easy to understand. It facilitates efficient development and maintenance by enabling different metadata editors to write in the same style. The standard template for column definitions uses the following structure:

> The <legacy name> ….
>
> **Choose all that apply:**
> was added in <extract cycle/processing year>.
> (It) has data through <extract cycle>.
> (It) is <short description: clear, concise, and easy to maintain>.
> It is reported on <Form #, Line #>. ((It is transferred to OR included on <Form #, Line #>) (notated <'notation'>).) ((The format is <word for number> character(s) OR numeric. OR It is reported in (positive, negative, or positive and negative) whole dollars OR dollars and cents).) Valid values (if known) are …. Values are (if valid not known) …. It is (zero, blank, null) if not present OR if not applicable. (Values <other than valid> also appear.) (See <related fields>.)

Metadata includes valid values, when known. The process of data profiling is more informative when valid values are known. The actual values identified as part of the data profiling can be compared to the valid values. This allows for validation of a specific data field using the data in that one field. Metadata can also enable cross-validation to what should be the same field on other data tables, and can include more complex business rules for validation and assessment.

LEVERAGING SAS TO MANAGE METADATA

SAS is leveraged by CDW staff to support multiple areas of metadata management. SAS enables CDW staff to import metadata, iteratively process changes to data, and update external databases. The original metadata are typically created in one or more Excel spreadsheets. SAS is used to import the spreadsheet metadata and perform iterative processing using a SAS macro. Within the macro, PROC SQL is used to execute SQL statements in Microsoft SQL Server to load, update, or delete data.

IMPORTING DATA

After metadata are initially created in Excel spreadsheets, SAS is used to import the spreadsheet data into a SAS dataset. The SAS Import Wizard is used from the SAS menu under File→ Import Data→ Connect to MS Excel. Using the SAS Import/Export facility, the name of the spreadsheet to open is specified, as well as the Sheet name. The WORK library is used to create a temporary SAS dataset. Finally, a file name is typically specified to save the PROC IMPORT statements created by the Import/Export wizard for future use. The saved PROC IMPORT statements for this process are shown below.

```
PROC IMPORT OUT = WORK.METADATA
     DATAFILE   = "D:\Metadata\Metadata.xls"
     DBMS       =EXCEL REPLACE;
      SHEET     ="Sheet1$";
      GETNAMES  =YES;
      MIXED     =NO;
      SCANTEXT  =YES;
      USEDATE   =YES;
      SCANTIME  =YES;
RUN;
```

The output from the PROC IMPORT statements is a temporary SAS dataset, WORK.METADATA. The DATAFILE option is the filename including the complete path to locate the Excel spreadsheet on the PC. The DBMS option specifies the type of data in this case an EXCEL spreadsheet. REPLACE overwrites an existing dataset. SHEET allows one to specify which sheet in a spreadsheet with multiple sheets. GETNAMES uses the column names from the spreadsheet as variable names in the SAS dataset. CDW staff will typically view the newly created SAS dataset prior to loading to make sure that the spreadsheet was clean enough to import properly. Empty rows or columns or other anomalies will require edits to the spreadsheet, after which PROC IMPORT is re-run.

DATABASE CONNECTIVITY

After column definitions and other attributes are imported from Excel spreadsheets into a SAS dataset, a SAS macro is used to process the metadata and update the Microsoft SQL Server database. Connectivity to SQL Server is established through the LIBNAME statement with the ODBC engine type through the statement below:

```
libname lib odbc dsn=<data source name> schema=<schema name> insertbuff=2048;
```

In this example, LIB is the library reference to SQL Server that will be used in DATA or PROC steps. ODBC is the engine type, and refers to the Open Database Connectivity protocol used to provide an interface between ODBC-compliant applications (e.g., SAS) and databases (e.g., Microsoft SQL Server). DSN is the data source name assigned to a given ODBC configuration. SCHEMA is an option that refers to specific groups or roles that have permissions to database resources. Finally, the INSERTBUFF option is used to change the default input buffer size for database inserts. In this case, the input buffer size is set to 2048 bytes.

MACROS FOR ITERATIVE PROCESSING

A SAS macro is used to load the column-level metadata into Microsoft SQL Server, which is the database engine used to query and display metadata on the CDW website. The macro used to load column-level metadata follows:

```
%macro load_column_metadata(input=);

data _null_;
  if 0 then set &input end=last nobs=count;
  call symput('n',left(put(count,12.)));
  stop;
run;

%do j=1 %to &n;

       data _null_;
       m=&j;
       set &input point=m;
       call symput('c1' ,left(put(table_id,6.)));
       call symput('c2' ,left(column_name));
       call symput('c3' ,left(put(has_nulls,1.)));
       call symput('c4' ,left(put(Is_primary_key,1.)));
       call symput('c5' ,left(put(has_lookup,1.)));
       call symput('c6' ,left(lookup_table));
       call symput('c7' ,left(column_long_name));
       call symput('c8' ,left(column_desc));
       call symput('c9' ,left(data_type));
       call symput('c10',left(data_type_user));
       call symput('c11',left(distribution_type));
       call symput('c12',left(put(has_frequency,1.)));
       call symput('c13',left(put(has_STATS,1.)));
       call symput('c14',left(put(has_maps,1.)));
       call symput('c15',left(put(has_trends,1.)));
       call symput('c16',left(range_type));
       call symput('c17',left(put(MinLength,2.)));
       call symput('c18',left(put(MaxLength,3.)));
       call symput('c19',left(put(first_year,4.)));
       call symput('c20',left(put(last_year,4.)));
       call symput('c21',left(put(last_update,8.)));
       call symput('c22',left(put(refresh_date,8.)));
stop;
run;

proc sql;
       insert into lib.dbo_columns
       set table_id      =&c1,
       column_name       ="&c2",
       has_nulls         =&c3,
       Is_primary_key    =&c4,
```

```
        has_lookup        =&c5,
        lookup_table      ="&c6",
        column_long_name  ="&c7",
        column_desc       ="&c8",
        data_type         ="&c9",
        data_type_user    ="&c10",
        distribution_type ="&c11",
        has_frequency     =&c12,
        has_STATS         =&c13,
        has_maps          =&c14,
        has_trends        =&c15,
        range_type        ="&c16",
        MinLength         =&c17,
        MaxLength         =&c18,
        first_year        =&c19,
        last_year         =&c20,
        last_update       =&c21,
        refresh_date      =&c22
        ;
quit;

%end;
options nomprint;
%mend load_column_metadata;

%load_column_metadata(input=metadata);
```

Each row in the SAS dataset contains metadata for a unique column in the SQL Server metadata database, each of which has up to 22 separate attributes. The macro is used to populate these attributes for each column, reading input values from the temporary SAS dataset and iteratively processing them for each column.

The macro is named load_column_metadata with the %MACRO statement and a single parameter (input=). The macro includes a DATA step where the NOBS option is used on the SET statement to count to the number of observations in the specified SAS dataset. CALL SYMPUT is used to place that number into the macro variable &n.

A second DATA step is used inside a macro DO loop with the index running from 1 to &n. The SET statement is used to read one record at a time from the SAS dataset. The POINT option specifies the observation corresponding to the current value of the DO loop index.

The macro iteratively processes one record at a time. CALL SYMPUT is used to create 22 macro variables that are populated with the following attributes for the column-level metadata: table_id, column_name, has_nulls, Is_primary_key, has_lookup, lookup_table, column_long_name, column_desc, data_type, data_type_user, distribution_type, has_frequency, has_STATS, has_maps, has_trends, range_type, MinLength, MaxLength, first_year, last_year, last_update, and refresh_date.

INSERTING COLUMNS

PROC SQL is used inside the macro to insert column definitions and other attributes into Microsoft SQL Server. The INSERT INTO statement reads 22 separate macro variables that are created from the temporary SAS dataset and writes them to Microsoft SQL Server. The SET statement within INSERT INTO points to the SQL server table through an ODBC library reference created through the LIBNAME statement. The default option NOMPRINT is used to not display SAS statements generated by macro execution. For debugging purposes, the option could be changed to MPRINT. After the macro is compiled, it is called passing the parameter of the SAS dataset (METADATA) generated by PROC IMPORT from the imported Excel metadata spreadsheet.

DELETING COLUMNS

PROC SQL is also used outside of the macro. For example, the PROC SQL statements below are often used to remove all metadata attributes associated with a particular database table, identified through the TABLE_ID variable, when those attributes need to be quickly replaced.

```
proc sql;
  delete from lib.dbo_columns
```

```
  where table_id = 3260;
quit;
```

UPDATING COLUMNS

PROC SQL is also used outside the macro when only a few columns or columns across tables require an update. An example follows:

```
proc sql;
  update lib.dbo_columns
    set
       data_type      = 'tinyint',
       range_type     = 'Positive',
       data_type_user = 'Numeric',
       refresh_date   = 20110715
  where column_name   ='PENALTY_AMT';
quit;
```

In this example, a column named PENALTY_AMT is updated across all tables with that column. This sample job could be used when data type standardization is applied to a field that was formerly typed as character.

MONITORING SAS LOGS

SAS logs are reviewed to identify problems that might occur during the load. This ensures the integrity of database transactions and provides the ability to quickly recover from errors.

VALIDATING RESULTS

A staging website is used to validate metadata after it is loaded in SQL Server and is reviewed after each update made through SAS. It can also be used to validate proper attribute values used by statistical profiling functions to avoid run-time errors. After results are validated, metadata are moved to production and are available for general use by IRS employees.

CONCLUSION

SAS enables CDW staff to efficiently deliver large-scale metadata to the IRS Research community as part of a broader data quality initiative. Column definitions and other attributes are imported from Excel spreadsheets, iteratively processed using SAS macros, and exported to Microsoft SQL Server. Once in SQL Server, metadata are seamlessly managed through DATA steps and procedures via the ODBC engine. This process ultimately produces metadata that are published on the CDW website in a usable format to help IRS researchers quickly search for and understand the meaning of data available for analysis.

REFERENCES

SAS/ACCESS 9.2 for Relational Databases Reference
SAS User's Guide: Basics 1982 Edition
SAS Guide to Macro Processing
SAS Guide to the SQL Procedure

ACKNOWLEDGEMENTS

Elizabeth Schreiber – DCSUG Secretary Review and Encouragement

CONTACT INFORMATION

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

Robin Rappaport
RAS:R:RDD:TRD
1111 Constitution Ave., NW
Washington, DC 20001
Work Phone: 202-874-0578
Email: Robin.Rappaport@irs.gov