

Transitioning to SAS® Enterprise Guide

Michael Todd, Nth Analytics, Princeton, NJ

ABSTRACT

What are pros and cons of implementing SAS Enterprise Guide on a department-wide basis? This paper discusses whether pharmaceutical companies can reduce development and validation costs with the power-assisted coding tools Enterprise Guide provides. Even though pharmaceutical companies have tremendous investments in traditional macro-based applications, this paper argues that the benefits of a validated code-generator like Enterprise Guide cannot be ignored. However, costs involving learning curve and technical process conversion have to be considered

INTRODUCTION

Enterprise Guide (EG) is a visually oriented SAS programming environment that enables programmers to create diagrams linking objects that EG turns into SAS code. The object-oriented tasks in EG write SQL-based code. By incorporating an automated post-processor, EG offers a way to standardize programming style.

Enterprise Guide is an object-oriented interface to the SAS system. The main interface is the process flow. Initially, it is a blank screen on which to arrange the objects. The programmer sets properties for objects, and the objects generate SAS code “behind the scenes”. “Code objects” can also be included, containing user-written custom code, and these can be linked to the built-in objects in a process flow.

Hemedinger (2005) and Dmitrienko (2005) have good introductory papers on EG. Also SAS has a free online Enterprise Guide training course at http://www.sas.com/apps/elearning/elearning_details.jsp?pubcode=59271. It is important to note that EG is designed to be run interactively by “power user”. In order to get it to run in batch, as one would expect in a traditional SAS programming environment, you have to resort to some “tricks”, which will be discussed below.

ENTERPRISE GUIDE AS A PROGRAMMING ENVIRONMENT

Enterprise Guide can also replace the traditional SAS programming environment in the pharmaceutical industry.

ADVANTAGES

There are many reasons to do so:

1. Enterprise Guide generates better and more reliable code than programmers produce manually
2. Enterprise Guide enforces a standard programming style
3. SAS programmers can be developers while end users run the code.
4. Using stored processes, the source code can be secured and hidden from the end users.
5. Enterprise Guide can be deployed worldwide, allowing users to connect to a central SAS repository and run SAS programs without having SAS installed on their systems.

DISADVANTAGES

However, there are many pitfalls to go along with the advantages. First, the introduction of a new programming environment would require substantial IT resources for installation, validation, and FDA 21 CFR Part 11 compliance. Specifically, the following steps would be required:

- Creation of a Validation Plan
- A User Requirement or Design Specification. These documents make writing the OQ/PQ much easier because it facilitates writing tests for each user/functional requirement. For Enterprise Guide, this might be accessing data, testing security, backing up data, or testing a SAS program.
- Creation and execution of Installation Qualification (IQ)
- Creation and execution of Operational Qualification (OQ), including Security aspects.
- Creation and execution of Performance Qualification (PQ)
- Final Validation Summary Report, which also summarizes all the activities from the IQ/OQ/PQ.

Secondly, as SAS users, we are used to a stable programming environment. This is one of the nice features of

using SAS in regulated setting. The FDA and pharmaceutical industry change slowly. While the rest of the computing world has undergone several revolutions, SAS code for tables, listings, and graphs circa 1990, or earlier, can still be used. Anyone who has, for example, tried to restore a backup from 15 years ago can appreciate how valuable this is. The chance of that hardware/software combination still working is remote.

SAS has made a tremendous commitment to backwards compatibility for SAS programs across versions. This has been a great benefit to the pharmaceutical industry. For example, there has been no need to archive the operating system, database, and SAS version along with the code. SAS programs from 15 years ago still run, with at most minimal modification, and the legacy datasets can easily be updated to the current version. In short, even with the tremendous changes in the computing environment since the 1990s, the migration path for SAS programs and datasets has been relatively smooth.

One of the main reasons is that environment was text-based, consisting of ASCII text files that can be opened with any basic text editor. With Enterprise Guide, we are clearly moving away from this paradigm. Instead, we have an environment that, while it offers much more power, will change from version to version without backwards compatibility. It is not possible to convert an EG 4 project to EG 3, and it will not be possible to back-convert an Enterprise Guide 7 project to Enterprise Guide 4 in the future. SAS products will become like Microsoft products in that sense. We will migrate forwards, and the forward migration will not always be smooth. For example, because they are binary files, you cannot automate the conversion of Enterprise Guide projects by running the projects through a batch-processor and applying global changes, like you can with text-based SAS programs.

We have had some painful experience with Enterprise Guide migration. Nth Analytics is SAS consulting company for the pharmaceutical industry in Princeton NJ. We have been using Enterprise Guide in production since 2005, starting with EG version 3, and then migrating to EG 4.1 in 2006. We discovered that EG projects do not always migrate cleanly, particularly large, complex process flows. In most cases, we were able to break up the process flows in EG 3, and then reassemble them in EG4.1 as a workaround. A few projects simply could not be migrated, and had to be recreated.

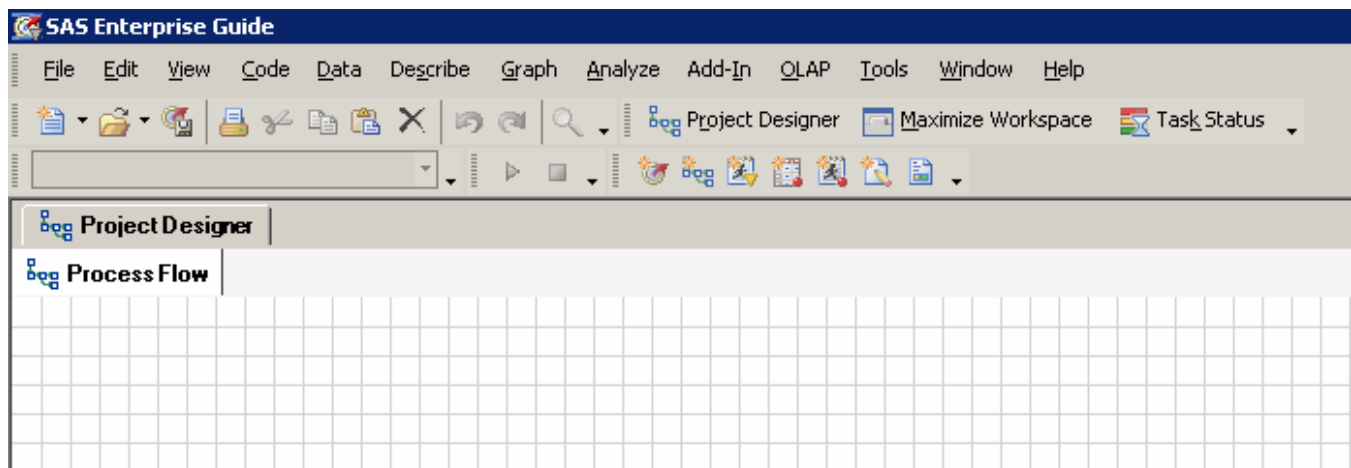
ARCHIVE COPY AS A SAFETY NET

Here is where the concept of an archive copy becomes critical. An archive copy in the Enterprise Guide environment is a SAS program that is generated from the code generated by the Enterprise Guide tasks. Regardless of what problems arise from version compatibility issues in the programming environment, the archive copy can always be run as traditional SAS programs. It is critical that the archive copies be generated from an automated, validated process. I describe such a process in the sections that follow.

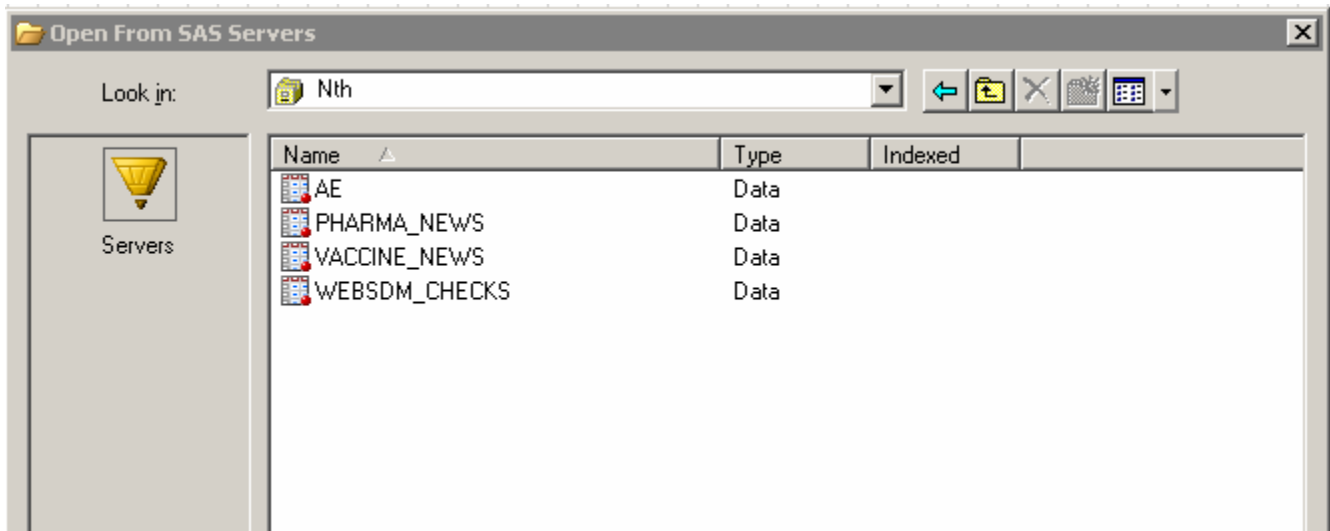
USING ENTERPRISE GUIDE

In this section, I examine EG as a code generator. In this sense, it's like the macro language. Macros expand out and generate SAS code; EG generates SAS code behind the scenes. Let's look at an example of EG doing a simple query.

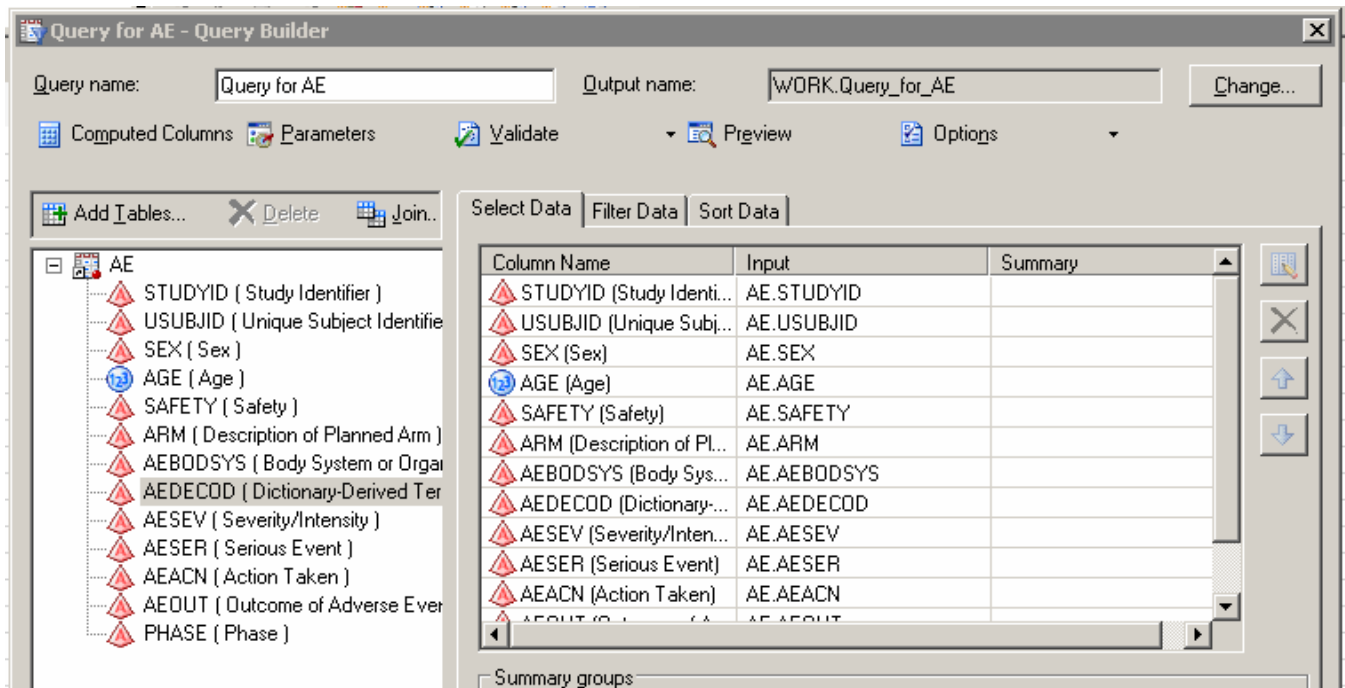
Start with a blank process flow:



Load the data from one of the libraries:



Build the query:



Build the select statement. One of the best things about EG is that allows you to pick valid values from an automatically generated list from the data.



Value	Formatted Value
ABDOMINAL ABSCESS	ABDOMINAL ABSCESS
ABDOMINAL DISCOMFORT	ABDOMINAL DISCOMFORT
ABDOMINAL DISTENSION	ABDOMINAL DISTENSION
ABDOMINAL PAIN	ABDOMINAL PAIN
ABDOMINAL PAIN LOWER	ABDOMINAL PAIN LOWER
ABDOMINAL PAIN UPPER	ABDOMINAL PAIN UPPER
ABDOMINAL TENDERNESS	ABDOMINAL TENDERNESS
ABNORMAL BEHAVIOUR	ABNORMAL BEHAVIOUR
ABNORMAL DREAMS	ABNORMAL DREAMS
ABNORMAL FAECES	ABNORMAL FAECES
ABORTION MISSED	ABORTION MISSED
ABORTION SPONTANEOUS	ABORTION SPONTANEOUS
ABSCESS	ABSCESS
ACCIDENT	ACCIDENT
ACCOMMODATION DISORDER	ACCOMMODATION DISORDER
ACIDOSIS HYPERCHLORAEMIC	ACIDOSIS HYPERCHLORAEMIC
ACNE	ACNE
ACRODERMATITIS	ACRODERMATITIS
ACTIVATED PARTIAL THROMBOPLASTIN TIME PROLONG...	ACTIVATED PARTIAL THROMBOPLASTIN TIME PR
ACTIVITIES OF DAILY LIVING IMPAIRED	ACTIVITIES OF DAILY LIVING IMPAIRED

It generates the PROC SQL SELECT statement for you:

Edit Filter

Column: AE.AEDECOD

Operator: In a list of values

Values:

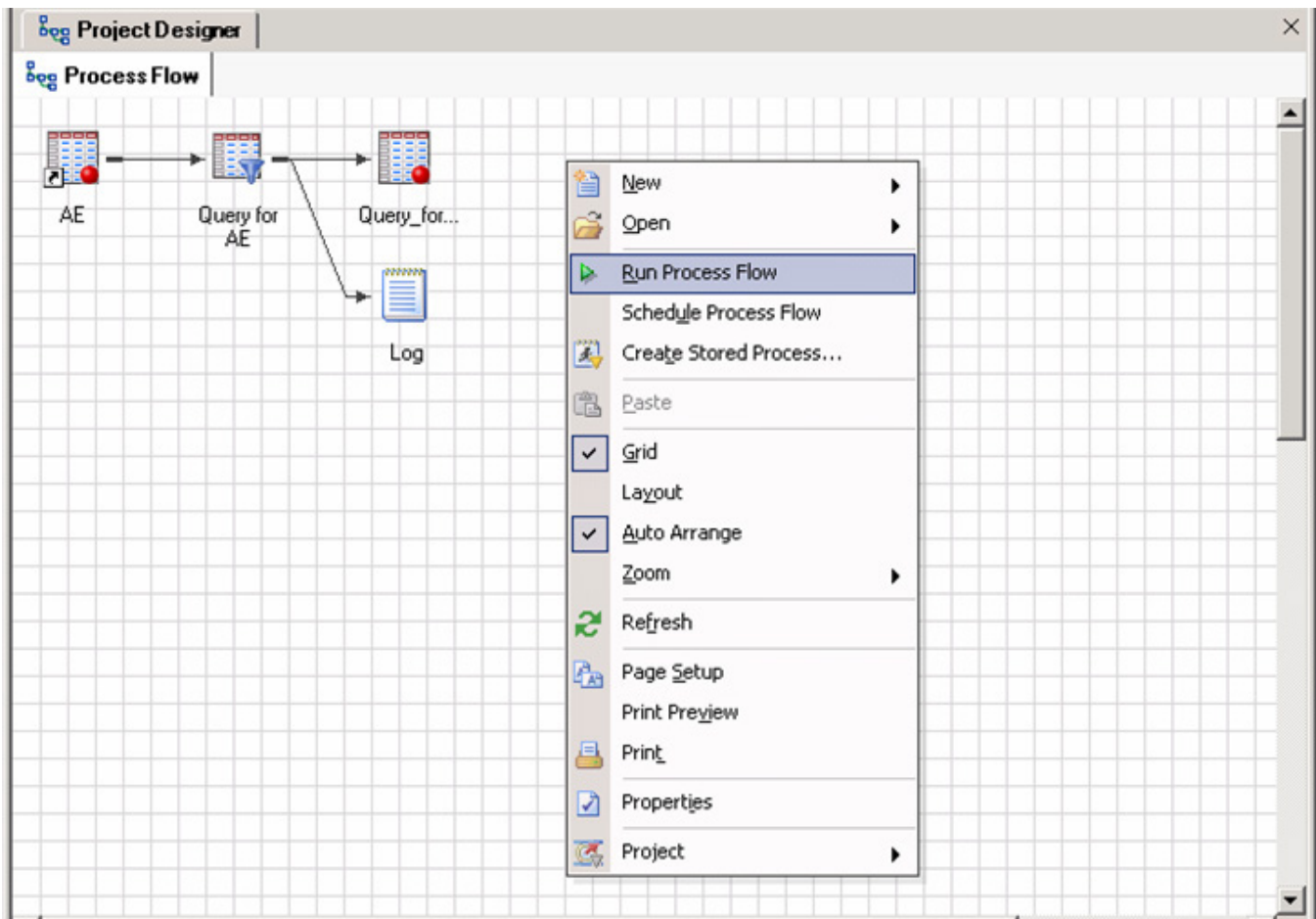
ABDOMINAL PAIN UPPER
 ABDOMINAL TENDERNESS
 ABNORMAL BEHAVIOUR
 ABNORMAL DREAMS
 ABNORMAL FAECES

Add
Remove

Filter definition

```
AE.AEDECOD IN ('ABDOMINAL PAIN UPPER','ABDOMINAL TENDERNESS','ABNORMAL BEHAVIOUR','ABNORMAL DREAMS','ABNORMAL FAECES','ABORTION MISSED','ABORTION SPONTANEOUS','ABSCESS','ACCIDENT','ACCOMMODATION DISORDER','ACIDOSIS HYPERCHLORAEMIC','ACNE','ACRODERMATITIS','ACTIVATED PARTIAL THROMBOPLASTIN TIME PROLONGED','ACTIVITIES OF DAILY LIVING IMPAIRED','ACUTE SINUSITIS','ACUTE TONSILLITIS','ADENOIDAL HYPERTROPHY','ADENOIDITIS','ADVERSE DRUG REACTION','AEROPHAGIA','AFFECT LABILITY','AFFECTIVE DISORDER','AGEUSIA','AGGRESSION','AGITATION','AGNOSIA','AKINESIA','ALANINE AMINOTRANSFERASE','ALANINE AMINOTRANSFERASE INCREASED','ALCOHOL INTOLERANCE','ALCOHOL POISONING','ALCOHOL WITHDRAWAL SYNDROME','ALLERGIC BRONCHITIS','ALLERGIC SINUSITIS','ALLERGY TO ARTHROPOD STING','ALLERGY TO VACCINE','ALOPECIA','ALOPECIA AREATA','ALTERED VISUAL DEPTH PERCEPTION','AMBLYOPIA','AMENORRHOEA','AMNESIA','ANAEMIA','ANAL INFLAMMATION','ANAL ULCER','ANGER','ANGINA PECTORIS','ANHEDONIA','ANHIDROSIS','ANIMAL BITE','ANIMAL SCRATCH','ANKLE FRACTURE','ANOREXIA','ANORGASMIA','ANOSMIA','ANTICONVULSANT DRUG LEVEL INCREASED','ANTICONVULSANT TOXICITY','ANTINUCLEAR ANTIBODY POSITIVE','ANXIETY','APATHY','APHASIA','APHONIA','APHTHOUS STOMATITIS','APPENDICITIS','APPETITE DISORDER','AREFLEXIA','ARRHYTHMIA','ARRHYTHMIA
```

Run the process flow:



The resulting query is written in PROC SQL. EG never generates datastep code. In a sense, by moving to EG, you are phasing out the datastep, except for special purpose “user interventions”.

CREATING AN ARCHIVE COPY

The process of generating an archive copy with EG is as follows:

1. Create the program using the process flow.
2. Select all objects and create a stored process
3. Run the stored process through a post processor

STORED PROCESSES IN ENTERPRISE GUIDE

EG will automatically create a stored process from the process flow. A stored process is a macro designed to be run interactively through a SAS client: web browsers, EG, and SAS add-in for Microsoft Office being the major clients. It is not designed to be run in batch mode, because the parameters have to be passed in at run time interactively. However, you can make it run in batch by building several stored processes off the same source code with different default parameters. For example, you could have an adverse event table with a single parameter: BYGROUP. You could then create separate stored processes

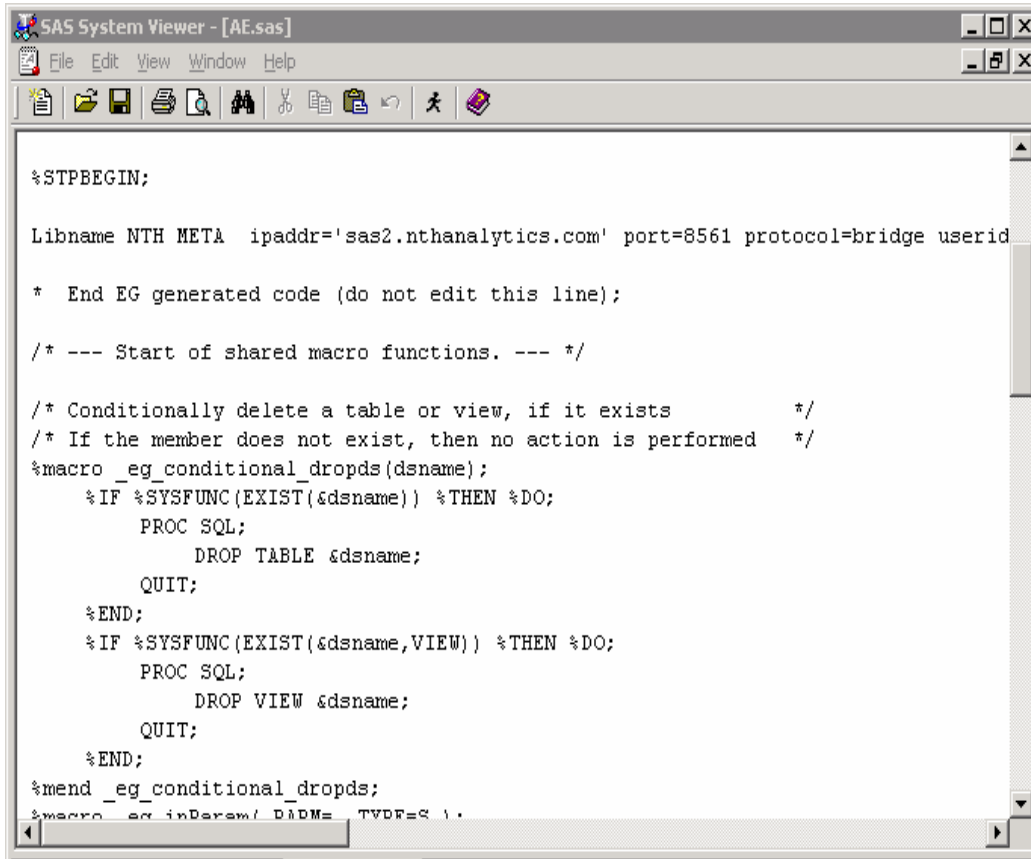
- AE_ALL: AEs with no subgroups (BYGROUP=)
- AE_SEX: AEs by sex
- AE_AGE: AEs by agegroup

To run them in batch, put objects for all three stored processes in the EG project. You need to create a VBScript program to batch the EG project, either from Windows Explorer or using the Windows scheduler. Both the output and the SAS log have to be directed to external files (otherwise they disappear when the project finishes).

STORED PROCESS SAS CODE

All stored processes have a source code file in the directory structure. It consists of EG code to make the stored process run, and normal SAS code. We want to get the normal SAS code into a standalone program.

This screenshot shows some EG-specific code:

The screenshot shows a window titled "SAS System Viewer - [AE.sas]". The window has a menu bar with "File", "Edit", "View", "Window", and "Help". Below the menu bar is a toolbar with various icons for file operations and execution. The main area of the window contains SAS code. The code starts with a macro call "%STPBEGIN;". It then defines a libname for "NTH META" with connection details. A comment line reads "* End EG generated code (do not edit this line);". This is followed by a comment "/* --- Start of shared macro functions. --- */". The code then defines two conditional macros: "%macro _eg_conditional_dropds(dsname);" which uses "%IF %SYSFUNC(EXIST(&dsname)) %THEN %DO;" to drop a table, and "%IF %SYSFUNC(EXIST(&dsname,VIEW)) %THEN %DO;" to drop a view. Both macros end with "%END;". The code concludes with "%mend _eg_conditional_dropds;" and the start of another macro "%macro _eg_inParam(DIDM= TVDF=S)".

```
%STPBEGIN;

Libname NTH META ipaddr='sas2.nthanalytics.com' port=8561 protocol=bridge userid

* End EG generated code (do not edit this line);

/* --- Start of shared macro functions. --- */

/* Conditionally delete a table or view, if it exists          */
/* If the member does not exist, then no action is performed  */
%macro _eg_conditional_dropds(dsname);
  %IF %SYSFUNC(EXIST(&dsname)) %THEN %DO;
    PROC SQL;
      DROP TABLE &dsname;
    QUIT;
  %END;
  %IF %SYSFUNC(EXIST(&dsname,VIEW)) %THEN %DO;
    PROC SQL;
      DROP VIEW &dsname;
    QUIT;
  %END;
%mend _eg_conditional_dropds;

%macro _eg_inParam( DIDM= TVDF=S )
```

Here is the normal SAS code:

```

SAS System Viewer - [AE.sas]
File Edit View Window Help

* End EG generated code (do not edit this line);

Libname NTH "\\Sas1\data\nth\data";

PROC SQL;
  CREATE TABLE WORK.Query_for_AE AS SELECT AE.STUDYID FORMAT=$15.,
    AE.USUBJID,
    AE.SEX,
    AE.AGE FORMAT=BEST22.,
    AE.SAFETY,
    AE.ARM,
    AE.AEBODSYS,
    AE.AEDECOD,
    AE.AESEV,
    AE.AESER,
    AE.AEACN,
    AE.AEOUT,
    AE.PHASE
  FROM NTH.AE AS AE
  WHERE AE.SAFETY = "YES" AND AE.AEDECOD IN ("AXILLARY PAIN", "BACK INJURY", "BACK PAIN"
" BENIGN PROSTATIC HYPERPLASIA", "BEREAVEMENT REACTION", "BLADDER DISORDER", "BLADDER SP
" BLOOD BILIRUBIN INCREASED", "BLOOD CALCIUM INCREASED", "BLOOD CHLORIDE DECREASED", "BL
" BLOOD LACTATE DEHYDROGENASE INCREASED", "BLOOD OSMOLARITY DECREASED", "BLOOD PHOSPHORU
" BLOOD TRIGLYCERIDES ABNORMAL", "BLOOD TRIGLYCERIDES INCREASED", "BLOOD UREA INCREASED"
" BOWEL SOUNDS ABNORMAL", "BRADYCARDIA", "BRADYKINESIA", "BRADYPHRENIA", "BRAIN NEOPLASM
" BREATH ODOUR", "BREATH SOUNDS ABNORMAL", "BRONCHIOLITIS", "BRONCHITIS", "BRONCHITIS AC
" BURSA DISORDER", "BURSITIS", "BUTTOCK PAIN", "CALCINOSIS", "CALCULUS URETERIC", "CALCU
" CARTILAGE INJURY", "CAT SCRATCH DISEASE", "CATARACT", "CATARACT OPERATION", "CATHETER
" CERVICOBRACHIAL SYNDROME", "CHALAZION", "CHANGE OF BOWEL HABIT", "CHEST DISCOMFORT");
QUIT;

```

POST-PROCESSOR

For either method, you need to clean up the generated code with a post-processor. Jemiolo (2002) describes the basic concept:

“The text file produced is all left aligned with a lack of hard returns between PROCs and DATA steps. To resolve these issues, a series of key words (i.e., PROC, DATA, RUN, and QUIT) are searched for. If these key words are encountered at the beginning of a line of text (not beginning with an “*”), a blank line is placed either before or after the line, as appropriate. Conversely, if these key words are not encountered in a line of text (not beginning with an “*”), then the line is indented by three spaces. It should be noted that each company has its own standards and tendencies and as a result, the list of key words should be changed appropriately to reflect these.”

For EG, the concept is similar; but the problem is simplified because the code coming out of the stored process is well structured to start with. However, there is a lot EG-specific code that has to be stripped out to make it look a like normal SAS program, as shown above.

LIMITATIONS OF ENTERPRISE GUIDE

For all advantages of EG, there are some significant limitations and disadvantages, aside from the learning curve, which can be fairly steep:

- Enterprise Guide has some unique features as a SAS programming environment. Therefore, SAS code from EG will not necessarily run in a SAS outside of EG. A testing step is required to verify that it does in fact run in batch or in the display manager. The major difference is that in EG, variable names can have spaces. For example, the following code is valid in EG:

```
PROC SQL;
  CREATE TABLE WORK.Query_for_AE AS
    SELECT AE.STUDYID FORMAT=$15.,
    AE.USUBJID,
    ('test') AS 'variable name with blanks'n
  FROM NTH.AE AS AE;
```

- Although EG has 80+ built-in tasks, it does not have a PROC REPORT task (and therefore does not have a built-in way to generate PROC REPORT code). You can write your own task as an add-in, but that's not easy and defeats the purpose of using an off-the-shelf product.
- There is no data step code generator in EG. Essentially, by going to EG, you are abandoning the datastep. I don't consider this a disadvantage. Datastep code can always be incorporated through code objects.
- Complex process flows can freeze up, and are unrecoverable in some instances.
- In a complicated process flow, tasks sometimes fail, forcing you to create an "ordered list", which usually solves the problem.

CONCLUSIONS

Despite some limitations, Enterprise Guide offers the evolution in SAS programming, and is an exciting replacement for the traditional Display Manager programming environment. However, the programming environment itself, not just the SAS code, will change from version to version. This is a major change in the SAS programming world, and has significant implications for regulatory compliance in the pharmaceutical industry. Using Enterprise Guide to automatically generate archive copies (i.e., traditional SAS code) of process flows mitigates these risks.

REFERENCES

Capturing SAS® Macro Code into an Executable SAS Program, David J. Jemiolo, DataCeutics, Inc., Pottstown, PA, 2002.

Boost Your Programming Productivity with SAS® Enterprise Guide Chris Hemedinger, SAS Institute Inc., Cary, NC SUGI 30, 2005.

SAS Enterprise Guide in Pharmaceutical Applications: Automated Analysis and Reporting, Alex Dmitrienko, Ph.D., Eli Lilly and Company, Indianapolis, IN, MWSUG, 2005.

ACKNOWLEDGMENTS

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.

CONTACT INFORMATION

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

Michael Todd
Nth Analytics
418 Wall St
Princeton NJ 08540
Work Phone: 908.672.5649
Fax: 253.595.7413
Email: info@nthanalytics.com
Web: www.nthanalytics.com