



# Using SAS® Enterprise Guide® to Create Standalone Programs

Nth Analytics

---

Mike Todd, President

[info@nthanalytics.com](mailto:info@nthanalytics.com)

908 672 5649

# What is a Standalone Program?

- All macro code and macro variables are resolved.
- Any input datasets and/or files are explicitly referenced. All libnames and external file references are in the program.
- No %includes: any referenced code is expanded and included in the program.
- Basic commenting. Detailed commenting for internal use should be stripped out.
- Simple standard header block, with program name, author, purpose, and date, noting it is the final, validated version. Internal header blocks with the revision history should be removed.

# Requirements

- To completely and accurately reproduce the outputs of the production system
- To be well formatted and easy to read.
- Stylistic consistency

# Standalone Program Process



Production Macro

```
%macro AE;  
...  
%mend AE;  
%AE
```

Production Dataset



AE

Raw Standalone Program

```
[JUNK]  
Libname out ...;  
PROC SQL; CREATE TABLE  
OUT.AE AS SELECT  
AE.STUDYID, AE.USUBJID,  
AE.AESEQ ...; ; QUIT;
```

Post-Processor

```
%macro postpr;  
...  
%mend postpr;  
%postpr
```

Final Standalone Program

```
/* std. header ...*/  
Libname out ...;  
  
PROC SQL;  
  CREATE TABLE OUT.AE AS  
  SELECT AE.STUDYID,  
         AE.USUBJID,  
         AE.AESEQ  
  ...;  
QUIT;
```

Test Dataset



AE

SAS log



PROC COMPARE



# MFILE METHOD

1. Run the macro using `OPTIONS MPRINT MFILE`.
  - Writes the expanded code from the macro to a file with all macro code and macro variables resolved.
2. Clean up the code with post-processor.
  - Includes addition of standard header block, basic commenting, indenting, white space, or any other standards to be enforced.

# Problem with MFILE Method

- Bad text splitting

```
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", "BACTERIAL
INFECTION", "BALANCE DISORDER", "BARTHOLIN'S CYST", "BASAL CELL CARCINOMA",
"BENIGN BREAST NEOPLASM", "BENIGN NEOPLASM OF BLADDER", "BENIGN PROSTATIC
HYPERPLASIA", "BEREAVEMENT REACTION", "BLADDER DISORDER", "BLADDER SPASM",
"BLEPHAROSPASM", "BLINDNESS TRANSIENT", "BLINDNESS UNILATERAL", "BLISTER", "BLOOD
ALKALINE PHOSPHATASE INCREASED", "BLOOD BICARBONATE DECREASED", "BLOOD BILIRUBIN
INCREASED", "BLOOD CALCIUM INCREASED",
"BLOOD CHLORIDE DECREASED");
QUIT;
```

# Inadvertent Resolution

- The MFILE method resolves all macro variables.
  - Including some that need to remain unresolved
- This can lead to embarrassing situations

# Preserving Macro Variables

```
proc report data=final &rptopts spacing=1;
  column label Stat
         sf_col
         ("Group|_" ("Elderly Adult" pcol1)
         ("  Adult" pcol2 ))
         ("Total All" pcol3 )
;
%divider;

define label/ " " display order=data width=54;
define stat/ display "Statistic" width=15 center;
define sf_col / display "Screening|Failure (N=&sf_nc)" width=10 center;
define pcol1 / display " (N=&t1)" width=15 left;
define pcol2 / display " (N=&t2)" width=15 left;
define pcol3 / display " (N=&t3)" width=15 left;
run;
```

Should resolve

Shouldn't resolve

# Oops!

```
proc report data=final nowd headskip split='|' headline missing spacing=1;
  column label Stat sf_col ("Group|__" ("Elderly Adult" pcol1) (" Adult" pcol2 ))
          |("Total All" pcol3 ) ;

  compute before _page_;
  line @ 1 149*' ' ;
  line @ 1 ' ' ;
  endcomp;

  define label/ " " display order=data width=54;
  define stat/ display "Statistic" width=15 center;
  define sf_col / display "Screening|Failure| (N=36)" width=10 center;
  define pcol1 / display " (N=20)" width=15 left;
  define pcol2 / display " (N=20)" width=15 left;
  define pcol3 / display " (N=40)" width=15 left;

run;
```

# Our client was not amused

- Their email

Just let you aware of the situation here. When Mike's group did all these programming, they just hand put in the population number when doing derived variables (they did not use macro variables). The problem with this is, when they corrected the primary immune pop, they forgot about other places where the hand put-in number should also need to be changed, so the update is not complete (Table 14.2.2.1.4/14.2.2.2.4).

- My response

We provided SAS code generated from our macros. Of course, we used macro variables.

However, during the process of generating the standalone code, some macro variables got resolved that were supposed to be left as macro variables. I'll search for any other such instances.

I'll fix this tonight and send [REDACTED] the revised code, plus a file compare showing the changes.

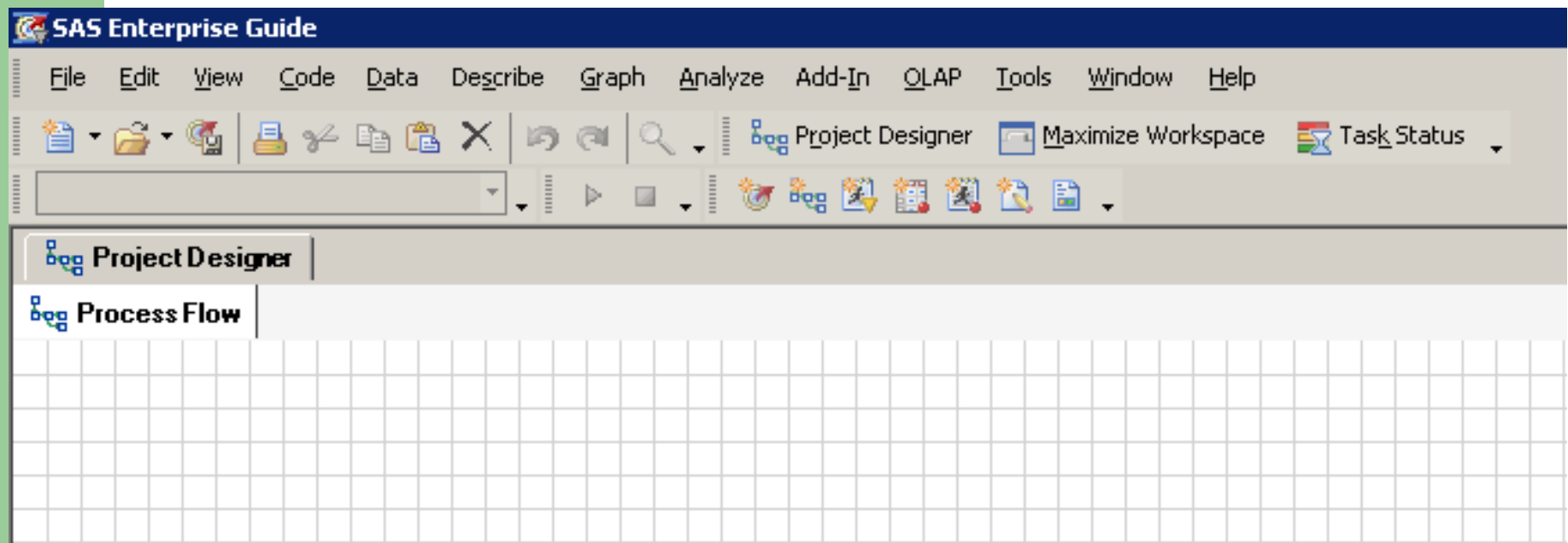
Sorry - my mistake!

# Enterprise Guide Method

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

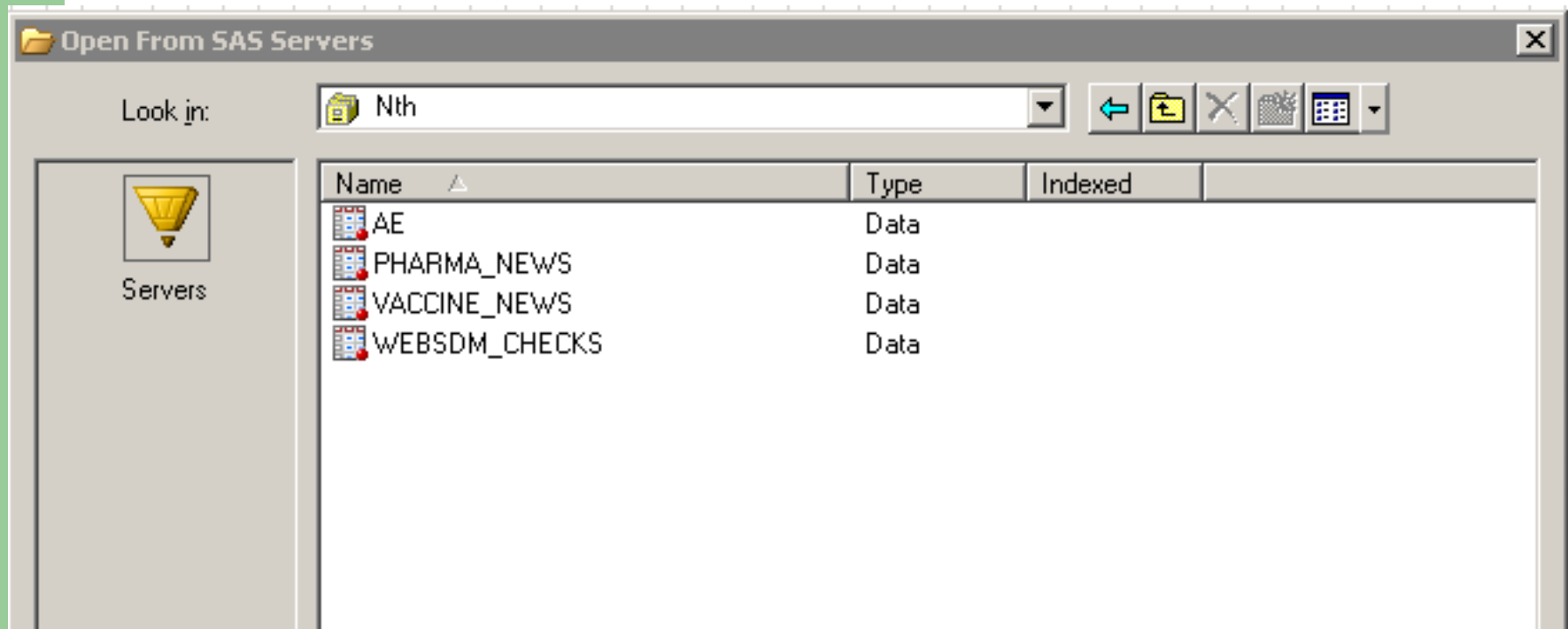
# Enterprise Guide

- Start with a blank process flow



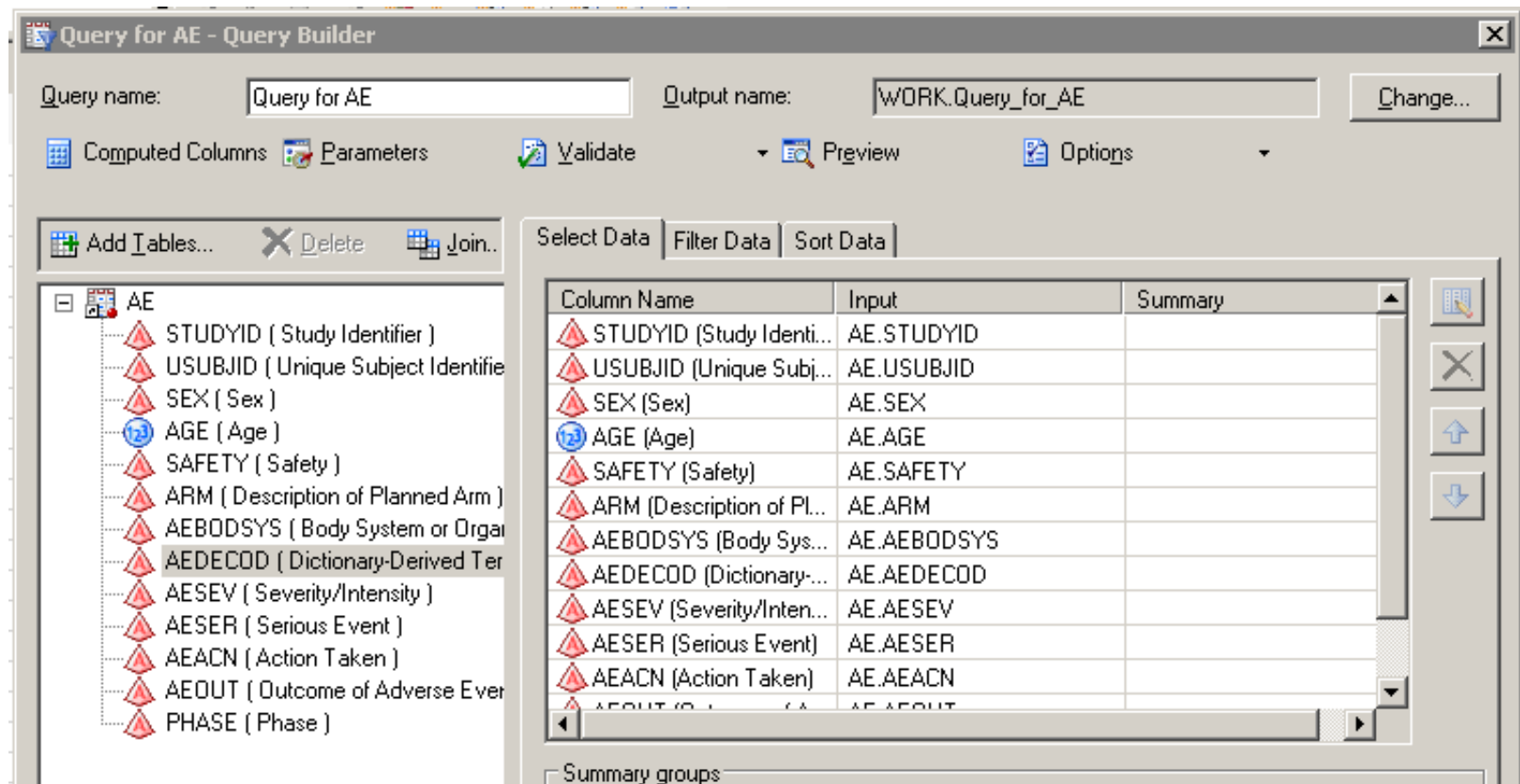
# Enterprise Guide

- Load the data from one of the libraries



# Enterprise Guide

- Build the query



Query name:  Output name:

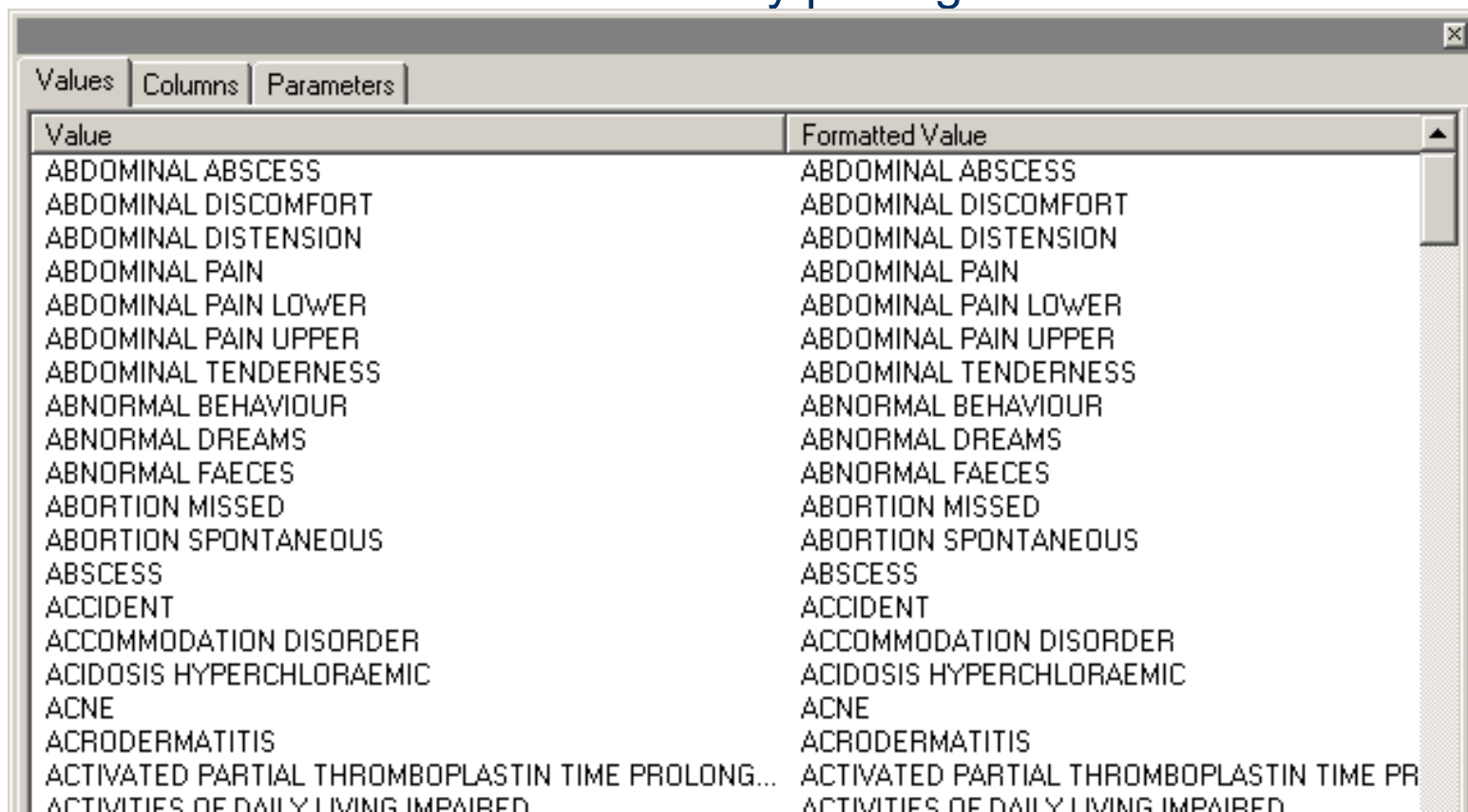
Select Data | Filter Data | Sort Data

Column Name	Input	Summary
STUDYID (Study Identifier)	AE.STUDYID	
USUBJID (Unique Subject Identifier)	AE.USUBJID	
SEX (Sex)	AE.SEX	
AGE (Age)	AE.AGE	
SAFETY (Safety)	AE.SAFETY	
ARM (Description of Planned Arm)	AE.ARM	
AEBODSYS (Body System or Organ System)	AE.AEBODSYS	
AEDECOD (Dictionary-Derived Term)	AE.AEDECOD	
AESEV (Severity/Intensity)	AE.AESEV	
AESER (Serious Event)	AE.AESER	
AEACN (Action Taken)	AE.AEACN	
AEOU (Outcome of Adverse Event)	AE.AEOU	
PHASE (Phase)	AE.PHASE	

Summary groups

# Enterprise Guide

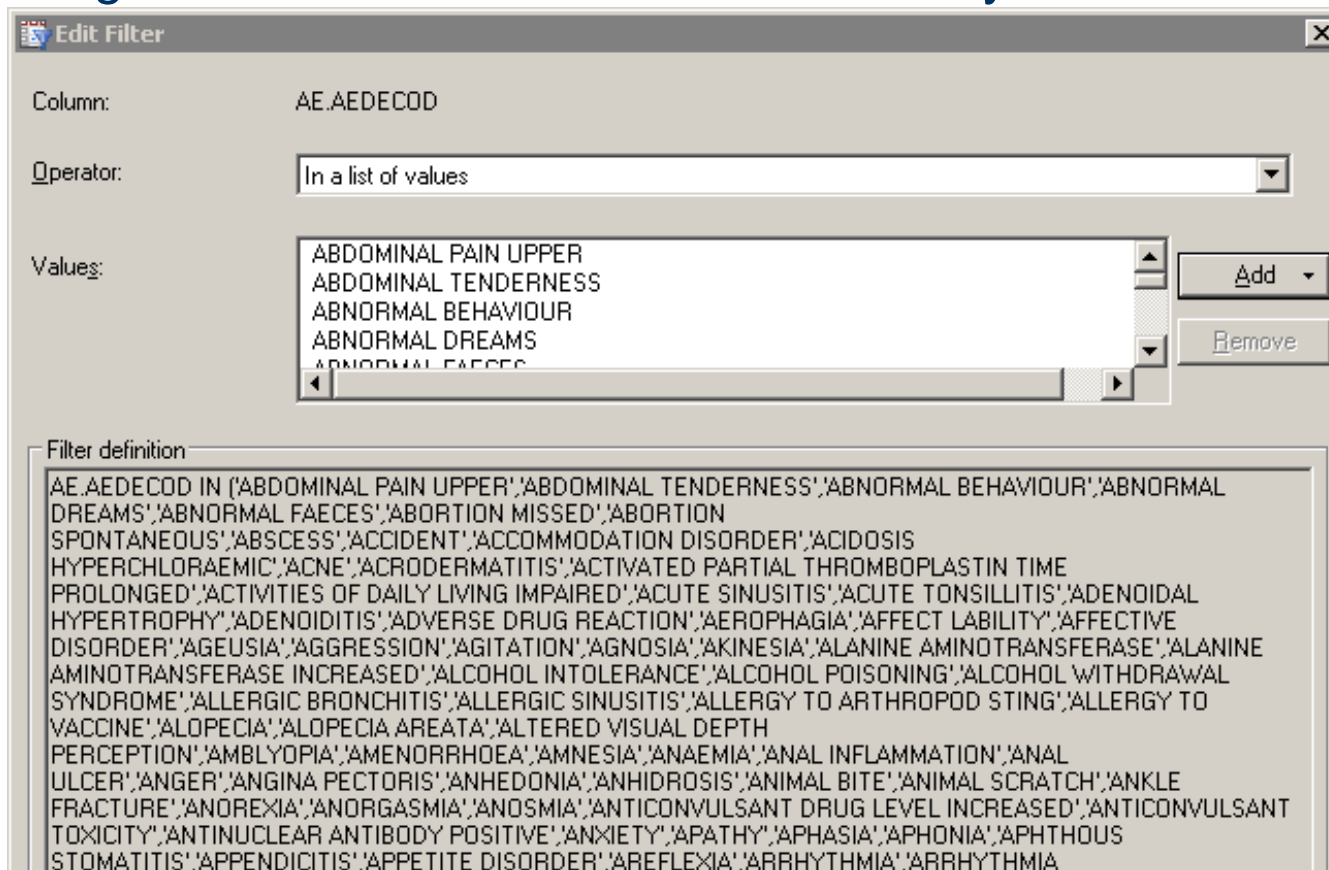
- Build the SELECT statement by picking values from a list



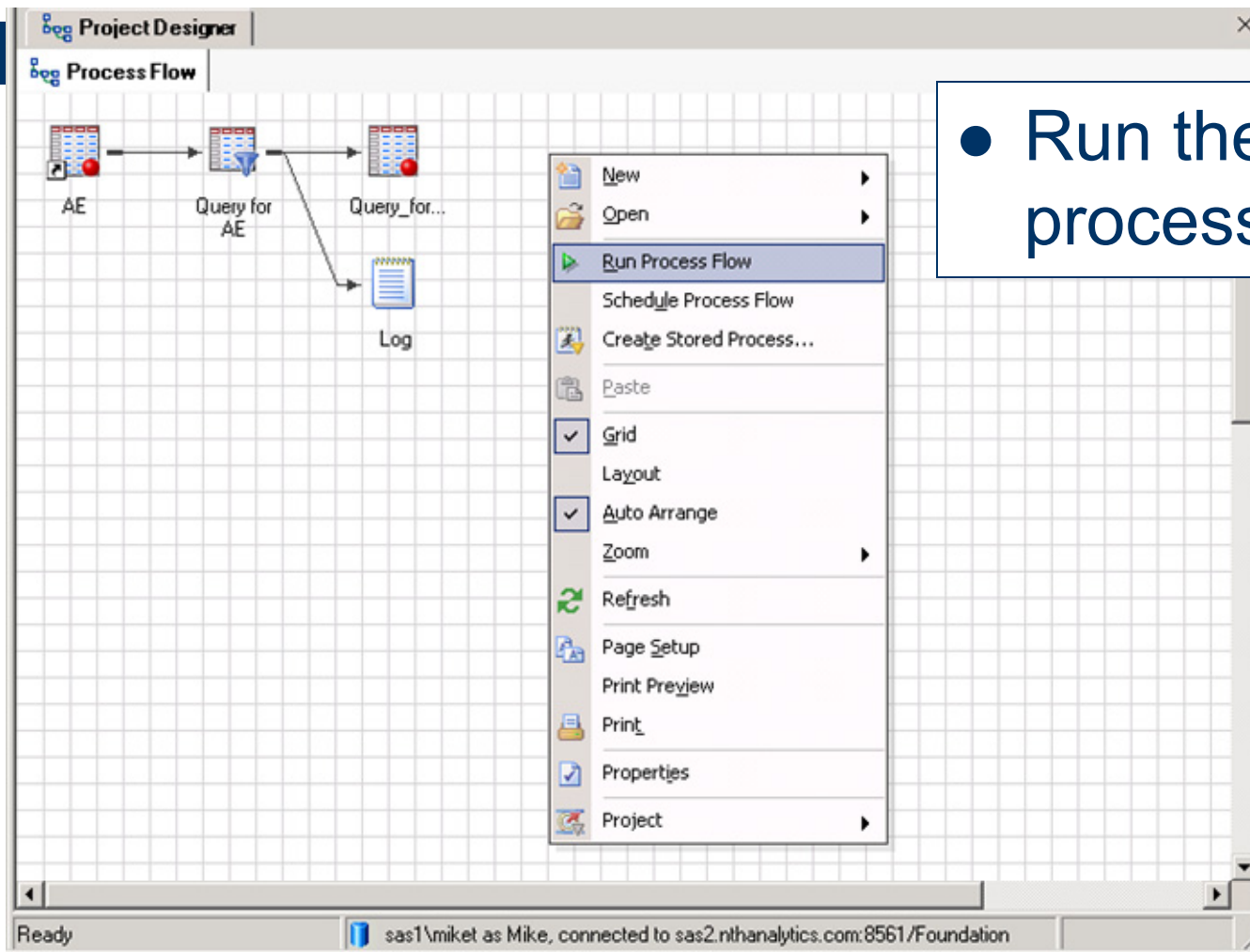
Value	Formatted Value
ABDOMINAL ABSCESS	ABDOMINAL ABSCESS
ABDOMINAL DISCOMFORT	ABDOMINAL DISCOMFORT
ABDOMINAL DISTENSION	ABDOMINAL DISTENSION
ABDOMINAL PAIN	ABDOMINAL PAIN
ABDOMINAL PAIN LOWER	ABDOMINAL PAIN LOWER
ABDOMINAL PAIN UPPER	ABDOMINAL PAIN UPPER
ABDOMINAL TENDERNESS	ABDOMINAL TENDERNESS
ABNORMAL 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

# Enterprise Guide

- EG generates the SELECT statement for you

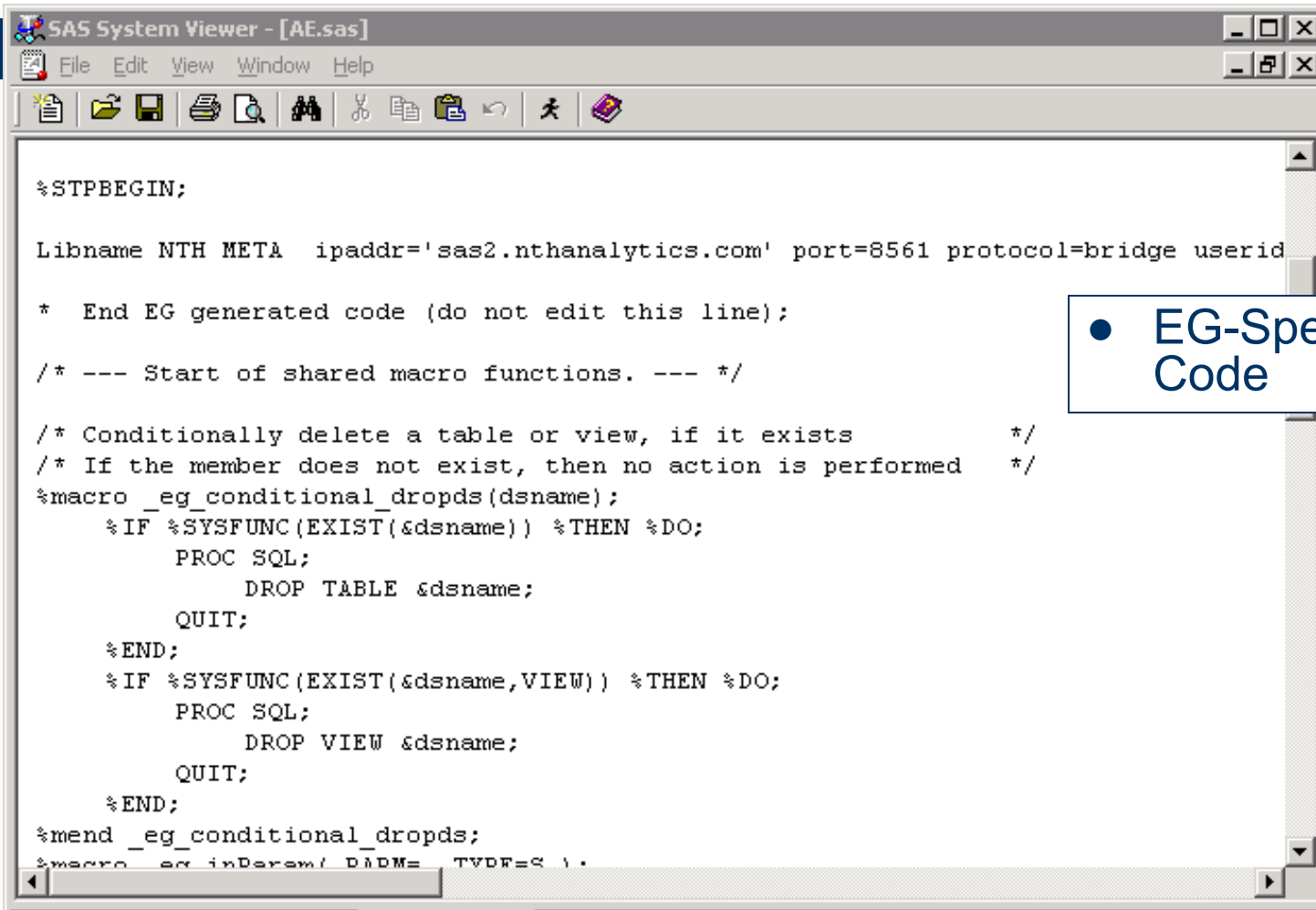


# Enterprise Guide



● Run the process flow

# EG/Stored Process SAS Code



```
SAS System Viewer - [AE.sas]
File Edit View Window Help
%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_drops(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_drops;
%macro _eg_inParam( DADMS TVDF=S );
```

- EG-Specific Code

# EG/Stored Process SAS Code

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

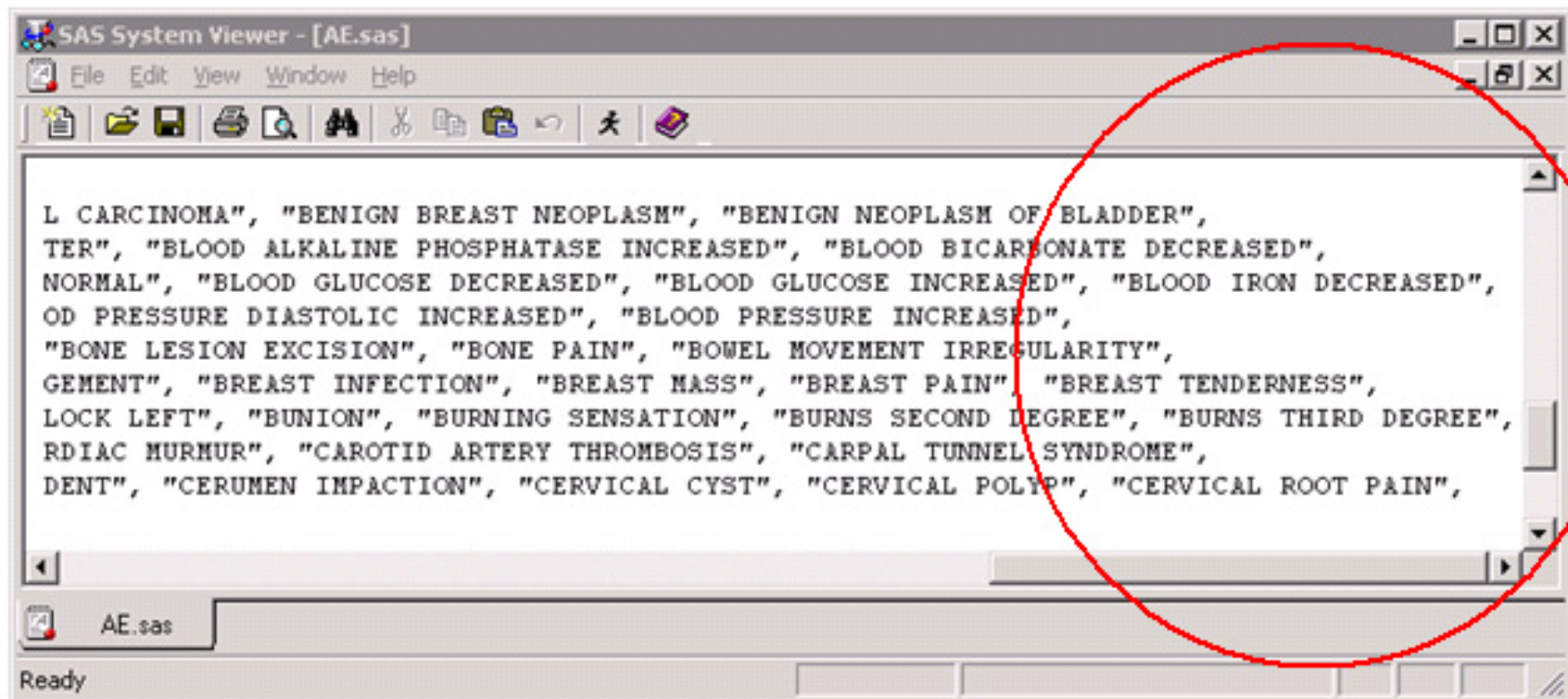
* End EG generated code (do not edit this line);

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;
```

- Normal SAS Code

# EG: Intelligent Text Splitting



SAS System Viewer - [AE.sas]

File Edit View Window Help

L CARCINOMA", "BENIGN BREAST NEOPLASM", "BENIGN NEOPLASM OF BLADDER",  
TER", "BLOOD ALKALINE PHOSPHATASE INCREASED", "BLOOD BICARBONATE DECREASED",  
NORMAL", "BLOOD GLUCOSE DECREASED", "BLOOD GLUCOSE INCREASED", "BLOOD IRON DECREASED",  
OD PRESSURE DIASTOLIC INCREASED", "BLOOD PRESSURE INCREASED",  
"BONE LESION EXCISION", "BONE PAIN", "BOWEL MOVEMENT IRREGULARITY",  
GEMENT", "BREAST INFECTION", "BREAST MASS", "BREAST PAIN", "BREAST TENDERNESS",  
LOCK LEFT", "BUNION", "BURNING SENSATION", "BURNS SECOND DEGREE", "BURNS THIRD DEGREE",  
RDIAC MURMUR", "CAROTID ARTERY THROMBOSIS", "CARPAL TUNNEL SYNDROME",  
DENT", "CERUMEN IMPACTION", "CERVICAL CYST", "CERVICAL POLYP", "CERVICAL ROOT PAIN",

AE.sas

Ready

## Limitations of EG

- Not really designed to run in batch
- Variables names with spaces are valid
- No PROC REPORT task
- No SQL set difference task
- Complex process flows can hang
- Objects don't always run in order