

Enumeration Technique for Efficient Clinical Laboratory Reports

Matthew J. Becker, STATPROBE, Inc., Ann Arbor, MI

ABSTRACT

The clinical laboratory reports may be a source of discouragement for any level of SAS® programmer. Conventional methods at STATPROBE, Inc. consisted of generating new programs for each report and project. The reasoning was that different projects contained different laboratory parameters. Therefore, a standard laboratory reporting macro would contain branching conditions for projects with different laboratory parameters.

To remedy this situation, a laboratory enumeration file is generated that is included by a standard laboratory reporting macro. The enumeration file consists of macro variables defining the laboratory parameter, a descriptive name, units, and the output format. Indexing the macro variables with increasing numerical values beginning at one (1) allows for macro do-loop processing in the standard laboratory report macro when generating the report.

STATPROBE established the enumeration technique to build efficiency in the reporting of clinical laboratory data. Using the enumeration file simplifies the clinical laboratory reporting. Coding the enumeration file and writing a program that invokes the desired laboratory reporting macro are the only requirements.

INTRODUCTION

In order to provide an efficient method of producing the clinical laboratory reports, STATPROBE developed the clinical laboratory enumeration technique. This technique provides programmers with a system to easily create clinical laboratory reports for different projects with varied laboratory parameters. The laboratory parameters desired on the clinical laboratory report distinguish different projects. Thus, the clinical laboratory report is similar in design across projects. Therefore, instead of re-programming a clinical laboratory report program to list different laboratory parameters, STATPROBE creates an enumeration file that defines the laboratory parameters to output for each project.

The enumeration macro defined for each project provides the vital information for the standard laboratory macro that generates the clinical laboratory reports. Enumerated macro variables constitute the enumeration macro. These variables define the column headings, units, laboratory variable name, and output formats.

CLINICAL LABORATORY ENUMERATION TECHNIQUE

Clinical laboratory reports are trivial in design. After standard columns, the remaining columns list the data for the laboratory parameters for a particular study. Therefore, one sees that we can create a list of the laboratory parameters

displayed on a clinical laboratory report table. However, one needs to be sure there is an "index" to separate laboratory variables. Therefore, a programmer must define the list of laboratory variables in the order they appear on the table. Furthermore, defining macro variables in a way that promotes macro do-looping makes the report programming trivial. This is the basis of the clinical laboratory enumeration technique.

ENUMERATED LABORATORY PARAMETERS

To enumerate the laboratory parameters, STATPROBE defines macro variables indexed with increasing numerical values beginning at one (1). Defining macro variables with a sequential numeric index allows for macro do-loop processing in the laboratory reporting macro.

As an example, assume a project has three laboratory parameters: hemoglobin, platelets, and creatinine with variable names HGB, PLATE, and CREAT respectively. Furthermore, the order of display on the clinical laboratory report is hemoglobin, platelets, and creatinine. Figure 1 lists the necessary enumeration file.

```
%Let var1=hgb;
%Let var2=plate;
%Let var3=creat;
%Let numvar=3;

%Let head1=%str(HEMOGLOBIN);
%Let head2=%str(PLATELETS);
%Let head3=%str(CREATININE);

%Let unit1=%str(g/DL);
%Let unit2=%str(1E3/mm3);
%Let unit3=%str(U/L);

%Let fmt1=5.1;
%Let fmt2=5.;
%Let fmt3=5.;
```

Figure 1. Example Enumeration File

The first group of macro variables (var1-var3) equal the laboratory parameter variable names in the order they appear on the table. Additionally, the macro variable numvar equals the number of variables output. Next, the macro variables head1-head3 equal the column heading text strings for the laboratory parameters. The units and output format specifications equal the laboratory parameter units and output size.

With enumerated macro variables like those above, it becomes trivial to output them across the columns of a table.

IMPLEMENTATION OF ENUMERATED FIELDS

The standard laboratory report macro implements the enumerated macro variables with a macro do-loop. First, the

enumeration macro for the specific project is included within the report macro. This initializes the macro variables for reference within the standard report macro.

Within the report macro, a macro do-loop is repeatedly called for the column headings, units, and the laboratory parameter and output format.

```
Put @&c1 "PATIENT NUMBER"
%Do j=1 %to &numvar;
  @&eval(&c&eval(&j+1)) "&&unit&j"
%End;
```

Figure 2. Do-Loop Process

Figure 2 contains a code segment that outputs the unit text macro variables across columns. Macro variables c1, c2, ..., c<n> equal the column number for each column. The first column contains the text "PATIENT NUMBER." Following the first column, the do-loop executes from 1 to &numvar. Reminder: The macro variable numvar contains the number of laboratory parameters output. The code @&eval(&c&eval(&j+1)) will execute as follows:

1. The code segment %eval(&j+1) resolves first. For j=1, the result is 2. At this time, the original code is @&eval(&c2).
2. Code segment %eval(&c2) resolves. This resolves to the column number stored in the macro variable c2.

The code segment %eval(&j+1) is entered in order to offset the column by one. Since the text value "PATIENT NUMBER" is desired in column one, the laboratory parameter data will start in column two. Therefore, the do-loop beginning at 1 is incremented to output the data in the column constructed by the do-loop index+1. Without incrementing the macro variable j to 2 within the %eval function, the first unit text macro variable overwrites the first column header "PATIENT NUMBER."

```
Body:
If first.ptno then put @&c1 ptno 2. @;
Put
%Do j=1 %to &numvar;
  @&eval(&c&eval(&j+1)) "&&var&j" &&fmt&j
%End;;
If last.ptno then put;
Return;
```

Figure 3. Code to Print Out Body of Report

Figure 3 contains a code segment that outputs the laboratory parameters across the columns. The code segment determining the column executes exactly the same as described above for the column headings. The only difference is the output. The macro variables &&var&j and &&fmt&j resolve to the laboratory parameter variable name and the output format defined for that parameter. The two macro variables resolve as described below.

- Initially, the code is &&var&j &&fmt&j.
- After the first pass, the && resolves to & and the &j resolves to the current value of j in the macro do-loop. Since it is the first pass, &j resolves to 1. Therefore, the code is now &var1 &fmt1.
- The next pass resolves &var1 to hct and &fmt1 to 5.1. Thus, the code resolves to @<column position> hct 5.1.

Figure 4 contains the code segment for a condensed version of the standard laboratory report macro.

```
%Macro donull;

Data _null_;
  Set lab end=eof;
  By ptno;
  File print ps=65 ls=180 n=ps line=line linesleft=1;
  If _n_=1 then link header;
  Link body;
  If eof then link footer;
Return;

Header:
  Put @&c1 %uline
  /
  %Do j=1 %to &numvar;
    @&eval(&c&eval(&j+1)) "&&head&j"
  %End;
  / @&c1 "PATIENT NUMBER"
  %Do j=1 %to &numvar;
    @&eval(&c&eval(&j+1)) "&&unit&j"
  %End;
  / @&c1 %uline;
Return;

Body:
If first.ptno then put @&c1 ptno 2. @;
Put
%Do j=1 %to &numvar;
  @&eval(&c&eval(&j+1)) "&&var&j" 5.1
%End;;
If last.ptno then put;
Return;

Footer:
  Put @&c1 %uline;
Return;

Run;

%Mend donull;
```

Figure 4. Standard Laboratory Report Macro

One can see that there are three macro do-loops that output the column text headings, the column units, and the lab variable and output format. These are the coding segments that reference the enumeration file data.

PLANNED ENHANCEMENTS

The first planned enhancement is to construct a data driven standard laboratory reporting macro. Currently, the standard macro assumes all projects output identical standard columns in the clinical laboratory listings (i.e., treatment group, laboratory date, study day, patient number). These columns precede the columns that display the clinical laboratory data. STATPROBE will construct the standard macro so that projects with different standard columns will not bring about the need to write a project-specific standard laboratory macro.

In addition, STATPROBE, Inc. will allow the standard macro to split the laboratory column heading macros between lines for those that are longer than the defined column width. Currently, the width of each column has to be long enough for the column text macro variable to fit.

CONCLUSION

Supplying programmers with a standard laboratory report macro to generate clinical laboratory listings produces efficient programming. However, different projects require the listing of various laboratory parameters. Thus, a standard report macro that only outputs a specific set of laboratory parameters is not complete to use for all projects. Using the enumeration technique, STATPROBE achieves this goal in its reporting of the clinical laboratory data.

ACKNOWLEDGEMENTS

SAS is a registered trademark of SAS Institute, Inc. in the USA and other countries.

* indicates USA registration.

AUTHOR'S ADDRESS

Matthew J. Becker
STATPROBE, Inc.
3885 Research Park Drive
Ann Arbor, MI 48108
(313) 769 - 5000 ext. 102

E-mail:

- 72700,1075@compuserve.com
- statprob@oeonline.com