Paper 249

Using SAS/FSP® Software for Ad hoc Reporting

By Marc Schlessel
SPS Software Services Inc.

Abstract

This paper is geared towards mainframe SAS® programmers who are looking for an easy solution of creating ad hoc reports from user input without the interaction from a SAS® programmer. The SAS/FSP® product is a great tool to use for these types of projects. A user could input some selection criteria into a screen, and receive a report based on the criteria entered. This can be done interactively or in batch mode. This paper will address creating these reports in batch mode, but the same code and concept can be applied interactively.

Introduction

I run into users all the time that request reports they would like to submit themselves, whenever they want, with a different set of selection criteria. By using SAS/FSP® software, this can be accomplished with little effort. If your user doesn't ask for this type of request, it's always a good idea to keep this idea in the back of your mind, especially if your user is constantly calling you asking you to run a particular report using different selection criteria. You may want to take a little extra time and develop a system where you would turn it over to the user and eliminate phone calls. Another plus to using SAS/FSP® to accomplish this task is that with clists you can create it so the user never sees the SAS® code. It's always a good idea not to let the user see any code. This will eliminate possible problems in the future.

Another option to accomplish the task of users running their own reports and changing the selection criteria within the report is to use %let's at the top of the code. This however is not as user friendly as using SAS/FSP® software, plus the user would have access to the SAS® code, which potentially can cause problems down the road. This paper does not address the %let option.

Information needed from the User

Before starting, information from the user would need to be gathered. The two main sets of information needed are:

1. What different selection criteria would the user be asking for?
2. What should the report(s) look like?
    - Sort Order
    - What variables should be shown on the report
    - Titles
    - Footnotes
    - Etc…

After gathering this information, you should be ready to begin building your system.

Setting up the SAS/FSP® Screen

When in interactive SAS®, first you'll need to allocate the dataset(s) where the inputted data and the screen that you will be creating will reside. After allocating the dataset(s) where you want the data stored, you will need to issue an fsedit command to start creating your SAS/FSP® screen.

Note: *The SAS® dataset must be created prior to editing the file. This includes setting up the variables to be used for input by the user. This is shown in the data step below.*

The commands could look something like this:

```
Libname tst 'mhs.sas.dataset' disp=old;
Data tst.file1;
      Length Lastname $25. City $25. statecod $2. Zip $5. Operand1-operand4 $2.;
Proc fsedit data=tst.file1 screen=tst.testscrn;
Run;
```

- The above code will allocate the dataset where the data inputted by the user will reside and where the screen attributes and formats will be stored. The disp is set to old since we will be writing to the dataset.
- The data step above creates the variables used for the selection criteria. This example uses the length statement.
- The Proc Fsedit statement above needs to include the screen= option to identify the screen being created by the programmer (this will be discussed later within this paper).

The following in figure 1 is what will appear after submitting the above statements.

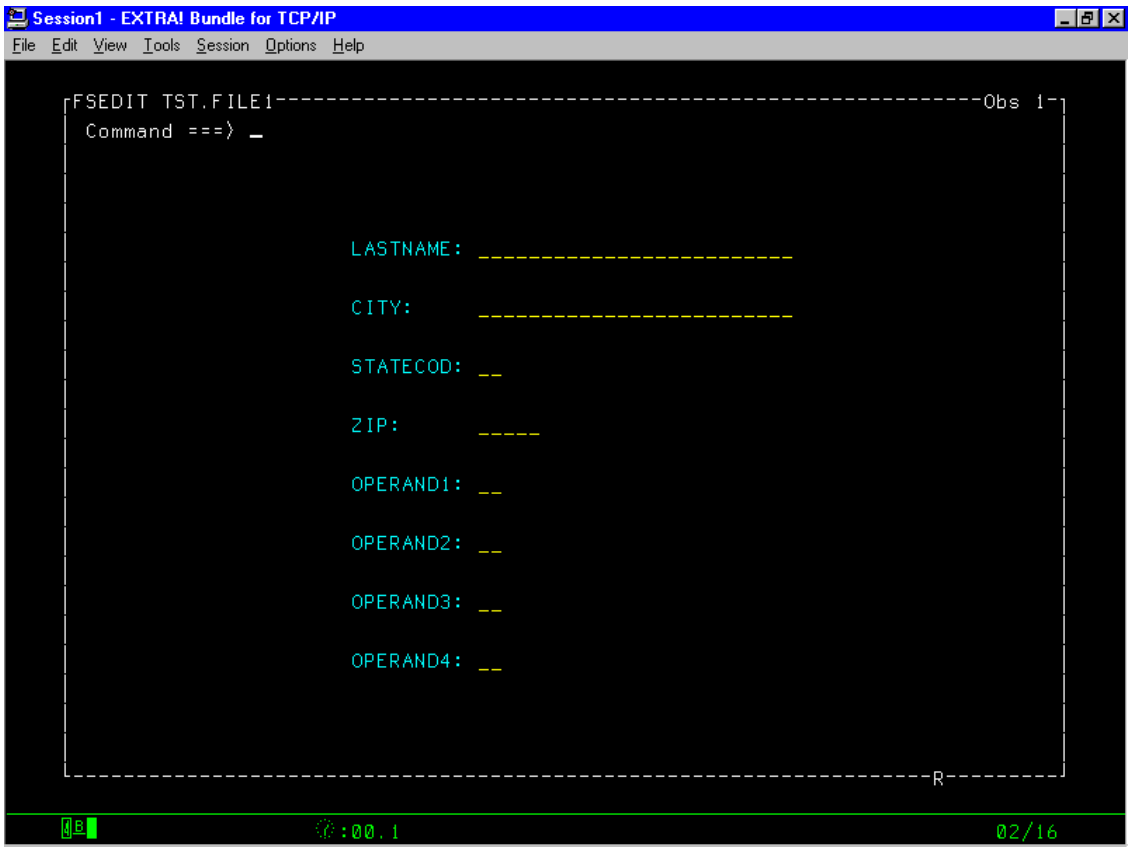

Figure 1

Next, type in “mod” on the command line to modify the screen (Figure 2). Choose option 2 (Screen Modification and Field Identification). After choosing option 2 the screen will look the same as in Figure 1 except it will say modify in the upper left corner. At this point you may modify the screen any way you want. The following bullets are a couple of helpful tools when using the modify screen:

- Typing ‘nums’ on the command line will insert line numbers. ‘nums off’ on the command line will get rid of the line numbers.
- Use the underscore (_) to identify all data fields.
- There must be a space between the data field and any other character.

When done modifying the screen, back out by hitting PF3 (End). At this time you will need to identify all the fields to the system by placing the cursor anywhere on the data field (_) that the system prompts you for. Next, choose selection 4 (Assign Special Attributes to Fields) in the mod menu (Figure 2). Here you will assign attributes to any or all of the fields you’ve defined. An example of a worthwhile attribute would be to change the max value of the operand fields to ZZ and the min value to AA. This will not allow a user to enter a equal sign (=) to the operand field. Hit PF3 (End) to get back to the mod menu (Figure 2). Now choose selection 5 (Modification of General Parameters).

Change the ‘Allow DELETE command’ and the ‘Allow ADD/DUP command’ values to ‘N’ (see figure 3). This will **not** allow the user to add, delete, or modify any observations in the SAS® dataset. PF3 (End) twice will get you back to the finished modified screen.

Figure 4 is an example of what a modified screen might look like.

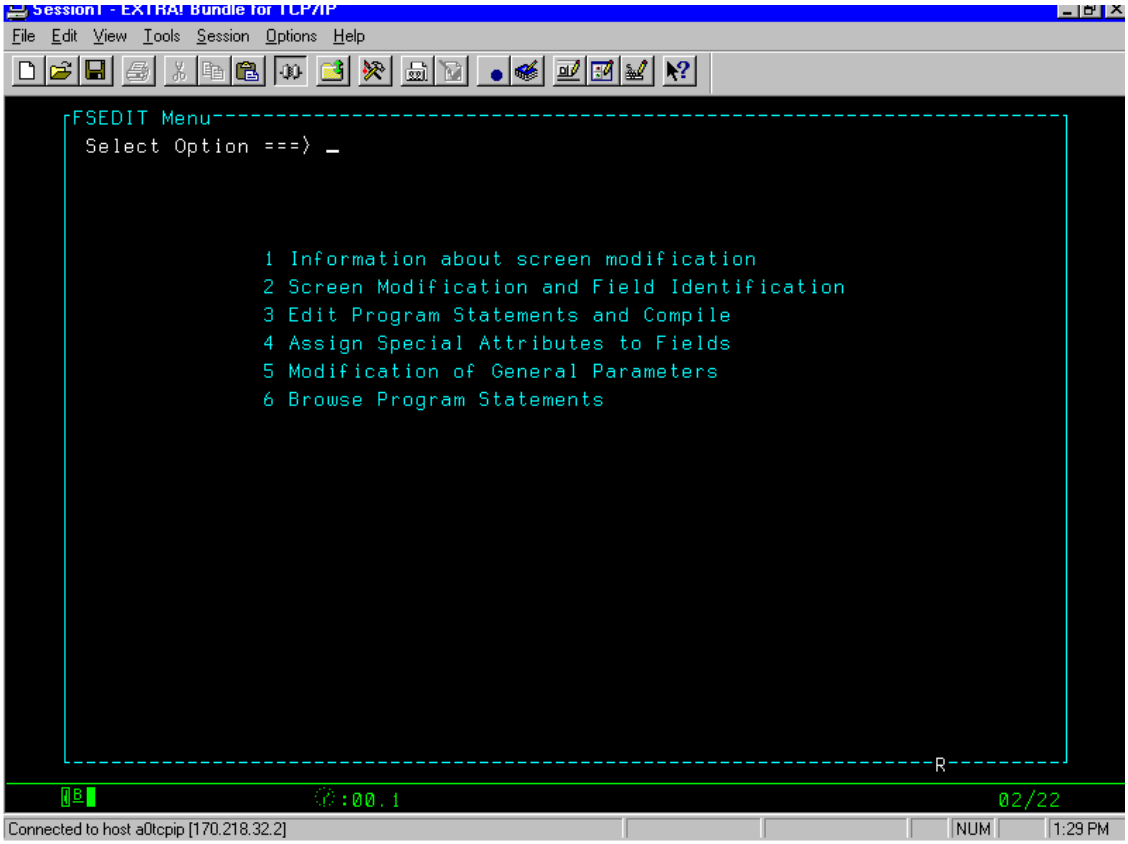

Figure 2

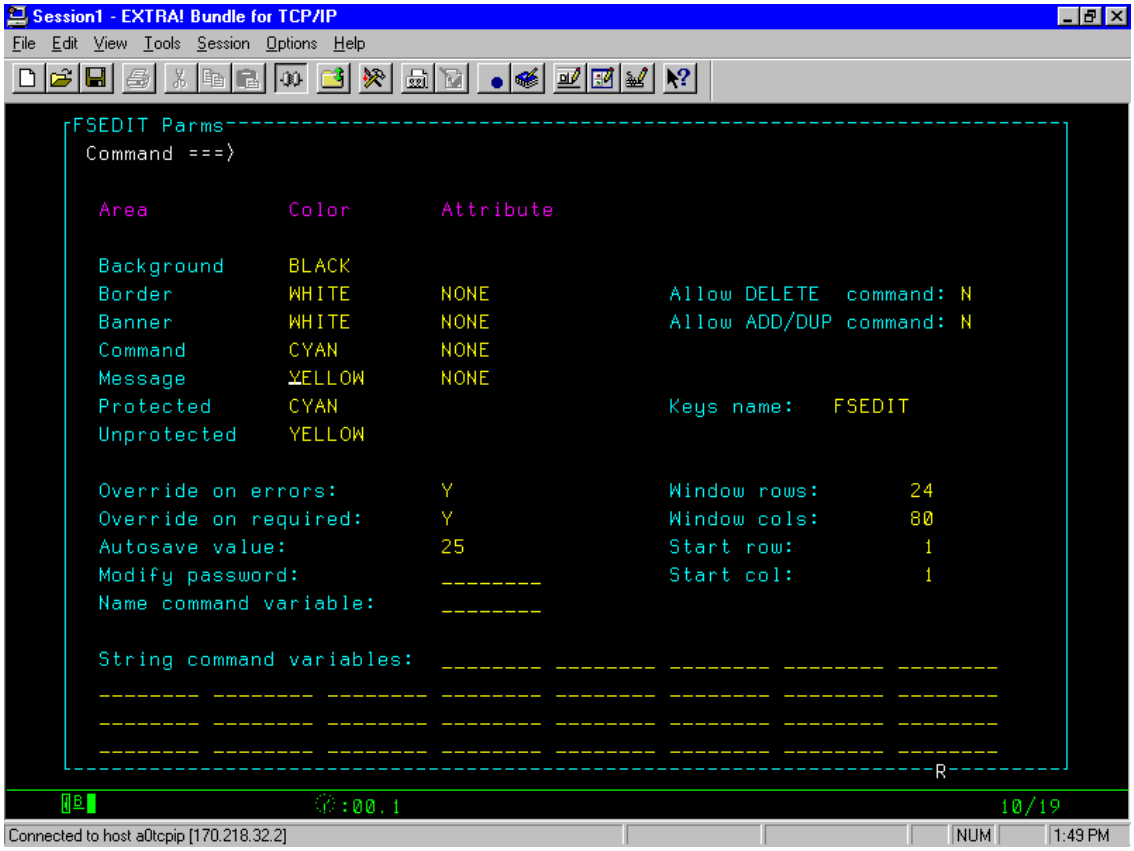

Figure 3

Figure 4

The Code

Since the data from the SAS/FSP® screen is already in a SAS® file, the code to make this work is quite simple. It would look something like the following:

The following is the test data used for this paper:

CITYNAM	LNAME	STATE	ZIPCODE
CLEVELAND	SMITH	OH	44115
CLEVELAND	ANDERSON	OH	44115
CLEVELAND	JACKSON	OH	44115
COLUMBUS	JACKSON	OH	44123
COLUMBUS	MARKS	OH	44123
NASHVILLE	ADAMS	TN	58941
NASHVILLE	GRAY	TN	58941
NASHVILLE	STEVENS	TN	58941

This test data is stored in a SAS® dataset called testdata

```
Data select;
        Set tst.testdata;
        If _n_ = 1 then do;
           Set tst.file1;
        End;
        Retain Lastname  City  statecod  Zip  Operand1-operand4;
        Array flds  lastname city zip statecod ;
        Array infl lname citynam zipcode state zipcode;
        Array op operand1-operand4;
        Do over flds;
            If flds ne ' ' then do;
                If op = 'EQ' then do;
                    If flds = infl;
                End;
                Else do;
                        If flds ne infl;
                End;
            End;
        End;
Run;
```

The above data statement is all that would be needed to create a SAS® file (in the above code it would be called 'select') of the data selected via the selection criteria from the SAS/FSP® screen.

Setting up Access for the User

There are many ways of setting up this type of system for a user to submit. One of the easiest ways and secure ways is to create clists. The following is an example of what a clist might look like to accomplish this task:

```
PROC 0

ATTN GOTO EX1
CONTROL NOMSG NOFLUSH PROMPT
SAS AUTOEXEC(”TMVS717.AGENT.CNTL(EXTABLE)”) +
OPTIONS(’NOSOURCE NONOTES’)

EX1: EXIT
```

The code would then reside in ‘TMVS717.AGENT.CNTL(EXTABLE). That code would look something like the following:

```
LIBNAME TST ‘MHS.SAS.DATASET’ DISP=OLD;

PROC FSEDIT DATA=TST.FILE1 SCREEN=TST.SCRNTEST;
RUN;

INSERT SAS CODE HERE IF NEEDED

ENDSAS;
```

The SAS® code above will take the user directly into the SAS/FSP® screen. The code that follows the FSEDIT statement (if you inserted any code where it says ‘*insert sas code here if needed’*) will run as soon as the user exits the screen. A proc print could be placed in here to bring a report directly back to the screen.

Conclusion

Lots of main frame SAS® programmers deal with this dilemma every day. Users like the control of running their own reports, whenever they want, and be able to change the selection criteria to meet their ever-changing needs. This paper addresses one way of many ways to accomplishing this dilemma. The SAS/FSP® product is a very easy product to use. It’s a good tool to use for Ad Hoc user generated reports. A system like this is a very easy way to “WOW” your users.