

Using REXX to Build a Front-end User Interface

Han-li Lee, United Illuminating Company

Abstract

This paper presents the UISSP (UI Social Support Programs) System, a menu-driven reporting system using IBM's REXX (Restructured EXtended eXecutor) language, base SAS[®], and SAS/FSP[®] (SAS version 5.18 under CMS operating system) for the United Illuminating Company, an electric utility company located in New Haven, Connecticut. The System uses the REXX to write a front-end user interface to the SAS System. It creates new files, updates existing files, and produces reports. This paper shows the techniques used for building such a user-friendly reporting system. It will also compare the use of REXX and SAS/AF[®] in building menu-driven systems.

Background

One of the United Illuminating (UI) Company's Corporate objectives is to be a good citizen in the communities the Company serves. This objective was established in 1986 and has been kept since. To achieve this objective, the Company is committed to assuming an appropriate share of the responsibility for the communities' growth and well-being and encouraging employees of the Company to participate in all types of civic activities.

The Company provides three type of services: volunteerism, in-kind contributions, and cash contributions. To document these services, a reporting system was needed. This project has been directed by Mr. Albert Harary, Vice President of Management Services Department and coordinated by Mr. Robert Buley, Manager of Financial Accounting and Reporting. The author has been assigned to build such a system. Since SAS has the capability and flexibility for creating user specified reports, it was selected as the language for the system.

In the beginning, SAS/FSP was used to build a screen for entering and updating the data. By using base SAS, four reports were devised for the three types of services respectively and a summary report for comparing the three types of services. And, these four reports provide the options to produce different, by organization or town, and by quarterly or up-to-date, reports. To use these options, the changes have to be made each time before running the system. This will

present a problem for those who do not have any knowledge or training in SAS language. A user-friendly system is thus required. Since back in early 1986 UI had not installed SAS/AF yet, the author then decided to use REXX to build a user-friendly front-end interface, UISSP system, to allow the user to enter the data and run reports easily.

Running the UISSP System

REXX is a general purpose, high-level language. This interface program is written in a REXX language EXEC, with a filename UISSP and a filetype EXEC. It contains CP and CMS commands. REXX programs are executed by an interpreter without first being translated into another form. This program is written under the host system CMS. To start running the system, the user just types a word "uissp," the filename of the EXEC, at the command line. The UISSP EXEC is then invoked. The attached Appendix of this paper shows an example of running the system by using certain options. Due to the limitation of the space for this paper, the screens are condensed, and the output listings are omitted. Below are the basic coding techniques used for building these screens. Four different techniques for preparing the reports are also presented.

Tips and Techniques

1. To create an EXEC file in REXX, the first line has to be a comment.

e.g.:/* comment */

2. To build a screen use: say " text "

e.g: see Appendix Screen 1.

```
say "- -----"
say "| Select an item:                                |"
say "|                                                                 |"
say "| 1. Create a new data set only                      |"
say "| 2. Update an existing data set only                |"
say "| 3. Produce a report                                |"
say "|                                                                 |"
say "| Enter selection (type 1 - 3, or QQ to exit)        |"
say "- -----"
```

3. To clear the screen each time before the designed screen shows up, use: `clrscrn`

4. Useful subroutines:

a. /* turn off cms error display */

`cmsoff:`

`signal off error`

`set cmstype ht`

`return`

b. /* turn on cms error display */

`cmson:`

`set cmstype rt`

`signal on error`

`return`

c. /* check answer for escape */

`get_answer:`

`parse UPPER pull answer`

`if answer = 'HX'] answer='QQ'`

`then exit`

`return`

5. Use the SELECT instruction, e.g.:

`SELECT`

`WHEN filename = 'REP1' then call report_1`

`WHEN filename = 'REP2' then call report_2`

`...`

`OTHERWISE NOP`

`END`

6. Four different ways to produce the reports. The first two SAS programs, REP1 SAS A and REP2 SAS A, are produced within the REXX EXEC, while the other two are produced externally and saved under different filetype and filemode as REP3 SSPSAS * and REP4 SSPSAS *.

a. use CMS command EXECIO with options FINIs and STring xxx... to create a SAS program for report 1, for example:

`execio 1 diskw rep1 sas a 1 f 80 "(string"`

`'OPTIONS DQUOTE NOTEXT82;'`

`execio 1 diskw rep1 sas a 2 f 80 "(string" 'DATA A; '`

`execio 1 diskw rep1 sas a 3 f 80 "(string"`

`' SET SSP.' name2';'`

`...`

`execio 1 diskw rep1 sas a 33 f 80 "(finis string" 'run;'`

b. use CMS command EXECIO with options FINIs and STEm xxxn... to create a SAS program for report 2, for example:

`Line.=`

`Line.1="OPTIONS DQUOTE NOTEXT82;"`

`Line.2="DATE A;"`

`Line.3=" SET SSP." name1";'`

`...`

`Line.33="H: PUT @51 'THE UNITED ILLUM CO'/"`

`...`

`Line.36="@55 'QUARTER ENDED " month year "' //"`

`...`

`Line.47="RUN;"`

`execio 47 diskw rep2 sas a 1 f 80 "(finis stem LINE."`

WARNING: Be sure to erase the SAS programs, REP1 SAS and REP2 SAS, before creating items a or b above to avoid stacking over to the old SAS program.

c. make a copy of the program rep3 sspsas and use CMS command EXECIO with options STring xxx to change the existing SAS program for report 3, for example:

`'copyfile rep3 sspsas * rep3 sas a (replace'`

`execio 1 diskw rep3 sas a 9 f 80 "(string"`

`' BY 'orgtown';'`

d. make a copy of the program rep4 sspsas and use instruction QUEUE in REXX to edit the report program, for example:

```
'copyfile REP4 SSPSAS * REP4 SAS A(replace'  
QUEUE 'COMMAND TOP'  
QUEUE 'COMMAND CHANGE ' / ' '  
      'NEW' / ' ' name1 / * *'  
QUEUE 'COMMAND FILE'  
'XEDIT REP4 SAS a (nopofile)'
```

REXX vs. SAS/AF

This UISSP system was built before the SAS/AF was installed at UI. SAS/AF together with base SAS is an application facility for creating a user-friendly front end to business applications, and it also is a great training tool. Both REXX and SAS/AF are easy to learn. However, it is also easy to make mistakes. The screens that are shown in the attached Appendix can be created in SAS/AF with MENU and PROGRAM screens. And the built-in facilities in SAS/AF provide much nicer looking screens and also make it easier to create a user-friendly front end to other products in the SAS System.

Conclusion

This paper presents an example of how to build a user-friendly interface to the SAS system in REXX. It is not the only, much less the best, way to accomplish the results. However, it does show some different uses of the techniques. The techniques employed in this paper can be useful in other applications. Although we now have powerful tools like SAS/AF, especially in version 6, to make the work easier, a REXX EXEC remains useful in many cases, and it does offer an alternative as a powerful interface building tool.

References

IBM VM/SP System Product Interpreter Reference,
SC24-5239-1

IBM VM/SP System Product Interpreter User's Guide, SC24-5238

IBM VM/SP CMS Command and Macro Reference,
SC19-6209-3

IBM VM/SP CMS User's Guide, SC19-6210-3

SAS, SAS/FSP and SAS/AF are registered trademarks of SAS Institute Inc. Cary NC USA

The author may be contacted at:

United Illuminating Company
80 Temple Street
New Haven, CT 06506
Telephone: (203)787-7904

APPENDIX RUNNING THE UISSP SYSTEM

Screen 1

Select an item:
 1. Create a new data set
 2. Update an existing data set
 3. Produce a report
 Enter selection (type number 1 - 3, or type QQ to exit)

2 (user-supplied value)

Screen 2

Please give any desired 1-6 character filename.
 (e.g.: Enter name, name is any desired filename,
 1-6 char, the file type is always SSP)

bbrev1 (user-supplied value)

Screen 3

If you are going to alter the data and save
 it under a new data set,
 please give a 1-6 character new filename.
 Otherwise, just press the ENTER.
 You may update the data when the screen shows up.
 Please wait!!!

bbrev2 (user-supplied value)

Screen 4

Date: 1986 (ddmonyy) Activity: V (C)ash, (I)n-kind, (V)olunteer
 UI Program: _____ Subscription: _ (do not enter)
 Organization: ANSONIA FIRE DEPT. KPA Code: _____
 Street Number: _____ Street: _____
 Town: ANSONIA State: CT Zip: _____
 type: 4 (Number 1 through 13)
 Contact: _____ Telephone: 787-7567
 Service Provided: FIREFIGHTER
 Last Name: BACLAWSKI First Name: EDWARD
 Length of loan from: _____ (ddmonyy) To: _____ (ddmonyy)
 Total Hours: _____
 Purpose: _____
 Amount or Estimated Value: _____ (9,999,999.99)
 Follow up Date: _____ (ddmonyy)

PF1=Help PF2=Return to SAS PF3=End PF5=Repeat PF6=Dup PF7=Backward
 PF8=Forward PF9=Add New Screen PF10=Left PF11=Right PF12=Command Line

Screen 5

Would you like to produce a report?
 Please answer (y or n)

y (user-supplied value)

Screen 6

Select a REPORT you would like to produce:

1. Volunteerism
2. In-Kind Contributions
3. Cash Contributions
4. Summary Report

Enter selection (type number 1 - 4, or type qq to exit)

1 (user-supplied value)

Screen 7

Quarterly report or Up-to-date report?

Answer (q or u)
or type QQ to exit

q (user-supplied value)

Screen 8

Please enter starting date: Month Year
The ending date is always today
(e.g.: September 1986)

september 1990 (user-supplied value)

Screen 9

Report by (ORG)anization, or by (TOWN) ?

Answer org, ORG, town or TOWN
or type QQ to exit

town (user-supplied value)

Screen 10

Would you like to view the listing (y/n) ?

If your answer is 'y'

When you finish viewing the listing, type 'quit' to exit

n (user-supplied value)

Screen 11

Which printer do you want to send your listing to?

Type:

- | | |
|---|---|
| p | for 3rd floor, 40 Temple |
| s | for 5th floor, 40 Temple |
| l | for 4th floor, 60 Temple |
| q | for 3rd floor, 80 Temple |
| d | for Computer Room, pick up the
listing at the Data Control, 80 Temple] |
| n | no print |

n (user-supplied value)