This article presents three commands, CBROWSE, CVIEW, and CEDIT, which can be used to browse, view, or edit datasets when the dataset name is on the screen. For example, while editing JCL, you may want to view a member of a Partitioned Data Set (PDS) which is used as a DSNAME in a DD statement. This article shows you how to set things up so you can do this by positioning the cursor on the dataset name and executing the cursor view command, CVIEW. In ISPF client/server mode you use just two mouse clicks: one to position the cursor and one on a function key pushbutton to execute the desired command.

The procedure to accomplish this will be described step-by-step. The first task is to create the CBROWSE, CVIEW, and CEDIT commands which are coded in REXX, and set up TSO so these commands can be executed from the command line. Next, assign the commands to function keys so they can be executed without having to enter the command on the command line. The various options described in Part III can be changed so that the function keys are displayed at the bottom of the screen (if they aren’t already there). The final step is to test these function keys to ensure that they work.

**THE REXX CODE**

The REXX code for the CBROWSE, CVIEW, and CEDIT commands is shown in Figure 1. The commands are implemented as edit macros, since they need to have access to the editor interface to determine the location of the cursor and to retrieve the dataset name from the screen. The original idea for these edit macros came from the article “Cursor Power for MVS/TSO: Using PDF Editor Interfaces” written by Andrew Brundell and published in the March 1992 issue of Enterprise Systems Journal.

Since the dataset name processing logic is the same for the three commands, all three edit macros can easily be created from the code in Figure 1. Near the beginning, an ISPF command is assigned to the variable “command”. Near the end, this variable is used to invoke the command against a dataset name which was retrieved from the data on the screen. The ISPF command which is executed can be changed simply by changing the value assigned to the variable “command” near the beginning of the code, as described by the comments.

The rest of the code reads a line of data from the screen, and, if the cursor is positioned on a dataset name, it retrieves the dataset name at the cursor position. It determines where the dataset name begins and ends by looking before and after the cursor for characters which are considered valid in dataset names. In addition to uppercase letters, numbers, and a few special characters, lowercase letters are also considered valid. The dataset name must be delimited by characters which are not valid in dataset names i.e., an equal sign ("="), comma (","), or space (" "). In particular, a closing parenthesis (""") will not be recognized as the end of a file name if other valid characters, such as letters or numbers, follow the closing parenthesis. As written, these edit macros will not handle relative member numbers of generation data groups, such as "DATA.SET.NAME(+1)". The code...
treats the plus sign ("+") as a delimiter, so the open parenthesis ("(") will be recognized as the end of the dataset name. Since the edit macro can’t read the catalog to find the current generation anyway, this isn’t much of a drawback, except for some misleading error messages. The comments provide more details on how the code works.

From the code provided, create three different members in a PDS. For REXX, the file type of the PDS should be EXEC. For one member, CBROWSE, use the BROWSE command as shown in Figure 1. In the second member, CVIEW, replace the BROWSE command with VIEW, and in the third member, CEDIT, replace the BROWSE command with EDIT. You now have three edit macros that execute different ISPF commands.

**In ISPF client/server mode you use just two mouse clicks: one to position the cursor and one on a function key pushbutton to execute the desired command.**

**EXECUTING THE MACROS**

To execute these macros from the ISPF command line, the library needs to be allocated to SYSEXEC or SYSUEXEC in a DD statement or in a TSO ALLOC command. If the PDS is allocated to SYSUEXEC, then the following TSO command must be executed to get it to search the user library:

```
ALTLIB ACTIVATE USER(EXEC)
```

If you execute other REXX commands or edit macros, this has probably been set up already. Once the library has been allocated and activated, the macros may be tested by executing them from the command line in an edit session. Try typing the name of one of the macros on the command line without pressing Enter. Then, type a dataset name on a data line in the main part of the edit screen. Move the cursor to the middle of the dataset name and then press the Enter key to execute the macro. The corresponding ISPF command should be executed using the dataset name where the cursor was placed.

**SETTING UP FUNCTION KEYS**

The next step is to assign the commands to function keys. I used the Keylist Utility described in Part III to assign the REXX commands to function keys as follows: CVIEW to F13, CBROWSE to F14, and CEDIT to F15.

The result is shown in Figure 2. The reason I separated CVIEW and CEDIT rather than placed them on adjacent function keys is in case I accidentally hit the wrong key or pushbutton. Due to their similarity, you might not notice if you had invoked VIEW when you had intended to be in EDIT, until you received the warning when exiting. Likewise, you might not notice being in EDIT rather than VIEW (which I use frequently to submit JCL with temporary changes) and wind up saving changes you didn’t want saved. On the other hand, you would certainly notice the difference between BROWSE and either VIEW or EDIT.

Next, issue the "FKA ON" command to display the function keys at the bottom of the screen. If you are using ISPF in client/server mode the function keys will show up as pushbuttons. If the function keys which have the commands do not show up, it may be necessary to change the range of keys displayed using the "Tailor GUI Function Key Definition Display" dialog shown in Figure 3. Since the commands are sensitive to the position of the cursor, having the commands assigned to function keys improves their usability; the commands may be executed without having to move the cursor to the command line to enter the command.

```
CLICK, CLICK

Now, you're ready to try it with the mouse. First, you must be using either EDIT or VIEW. You can't start from BROWSE because edit macros cannot be executed from BROWSE, and in client/server mode you can't position the cursor in the middle of the data being displayed. Therefore, if you are using ISPF option 1 (view/browse), make sure you have not selected browse mode, or use option 2 (edit) instead. The data you are editing or viewing must contain a dataset name or a member name. MVS JCL jobs will usually contain dataset names, while COBOL programs which reference copy books will contain member names. If the target is a member name, it must be a member of the same dataset as the member being displayed.

Here's how to browse, view, or edit a dataset or member with just two mouse clicks: Click the mouse button once on the dataset name or member name in the data on the screen. Then, click the mouse button a second time on the cursor browse, cursor view, or cursor edit pushbutton. The next thing you see on the screen should be the dataset or member you selected. You can click one more time on the EXIT pushbutton, to return.

FURTHER EXPLORING ISPF VERSION 4'S CLIENT/SERVER GUI

Hopefully, these articles have provided enough information for you to begin taking advantage of ISPF Version 4's client/server GUI. The majority of these articles have focused on windows and pushbuttons. However, there are other features, such as the menu bar, which are worth exploring further. I encourage you to continue your exploration and exploitation of the new features of ISPF.

The programs and text of this series can be obtained from the TECHSUPT LIB of NaSCOM, NaSPA's BBS, as filename SIMPSON.ZIP.

NaSPA member Robert Simpson has more than 16 years of computing experience, specializing in systems software support. He is experienced in installing and supporting OS/2 and related communications software, as well as data base and communications software on the MVS/ESA platform. He can be reached via CompuServe ID 71520,737 or Internet address 71520.737@compuserve.com.

©1996 Technical Enterprises, Inc. Reprinted with permission of Technical Support magazine. For subscription information, email membership@nasp.net or call 414-768-8000, Ext. 116.