Table of Contents

Table of Contents

Table of Contents

The MVS Workbench and Files Organizer Tool Package User's Guide Version 7.6

Nicholas L. Sun

MVS Programming Consultant
Poughkeepsie, NY. 12603, U.S.A.
nicksun@attglobal.net

December 7, 2001

1.0 Introduction

The **MVS Workbench and Files Organizer** is also called the MVS Project Work Manager tool package. It contains many extended and advanced ISPF command functions and various kinds of utility program functions with the interface of the ISPF system on MVS. It also contains a very powerful User−Exit command function that can allow you to define your own TSO/ISPF commands and Edit Macro commands.

This tool can be used as a personalized MVS project or work item files organizer and it can be customized to fit for your own needs. It can save you tremendous amount of precious time on searching for the data set names and conserve you a lot of energy on typing various kinds of long and tedious data set names on the ISPF panels.

This tool is very suitable for the MVS users who need to handle the code development or problem solving of many programs of several projects simultaneously. It can assist you to quickly identify which are the program files that were worked by someone on a project and now you are assigned to work on fixing the problems or program enhancement.

Many magnificent new features that you may not have seen in any other tools before have been provided by this tool package. For example, this tool package has provided a REXX Source Code Formatter function which can help you to write very neat REXX programs on the MVS system in very fast speed. It has provided a Desktop Rolling Sheet Scientific Calculator function which contains five very useful calculation modes that cover the Decimal, Hexadecimal, Binary, Date, and Time calculation functions. It has also provided a Desktop Monthly Calendar with a Things−To−Do function and a Notebook function which can be served as the weekly or monthly reminders and it can help you to very easily setup the project work schedules. By using the Notebook function, you can write down your notes, appointments, addresses, and phone numbers, etc. You can almost write anything about your project work that you need to save for the future reference.

This tool package has provided five process panels for the MVS project files organization in each Project Work Manager tool session. It can remember the project name, the data set name, and the PDS member file name that has been used previously. It can setup the special markers on the panels for the 'last used' project and the 'last used' data set, and it can re−display exactly the same screen of the data set name list panel and the PDS member name list panel that was last displayed before it was switched to another screen. With this special feature, you may repeatedly work on the same files very easily. It has also provided a special command, i.e. the '//' command, that can be used to switch between any two projects or work items back and forth very easily.

The five process panels can constitute a single Project Work Manager tool session. In addition to the five process panels, this tool package has also provided a special process panel for the Project Work Director function. This Project Work Director is one level higher than the Project Work Manager. You may enter a 'DIR' command on any of the five panels to enter into the Project Work Director session. On the Project Work Director panel, you can create many Project List Group entries in a database file and access your own primary and secondary Project List files as well as the Project List files of your team. It is a very powerful tool for the entire MVS Software Development organization to use.

On the process panels, this tool has provided the 'Extended String Search' and 'Massive Change' functions using the Edit Ring concept, which can assist the MVS programmers and testers to greatly improve their work performance and productivity.

This tool has simplified the MVS file transmit function. It has provided four different send file commands, i.e. the 'PRTPDS', 'SENDFILE', 'SENDPDS', and 'SFX2MVS' commands, that can help the users to send the MVS files to a target destination very easily. When you send a MVS file by using this tool package, you don't need to worry about whether the file you send is a PDS or a sequential file because the 'SENDFILE' command can take care of it for you. The 'SENDFILE' command provided by this tool is so flexible that it can even be

used to send just a code segment in the edited file to the target destination.

This tool package was written in REXX. It is a very easy−to−learn and easy−to−use tool. Some sections of this Reference Guide were extracted from the on−line tutorial guide of this tool package. After you finish reading this document, you may skip reading those on−line tutorials as you wish. However, please keep in mind that not all of the on−line tutorials provided by this tool have been covered in this document and some of the detail descriptions documented in this Reference Guide are not included in the on−line tutorials either.

The file access methods provided by this tool package is not only very unique but also very creative. It is a very extraordinary tool package with various kinds of ingenious design which can revolutionize the traditional ways for the MVS programmers and testers to handle the MVS files in a very much easier and efficient fashion. By using this tool, the users can save tremendous amount of precious time.

In summary, the following are the highlights of the essential features and special functions provided by this tool package:

1. Manage the MVS files through the project and work item database files.
2. Organize the MVS project or work item files using the newest file access method.
3. Eliminate the tedious typing work of the long data set names on the ISPF panels.
4. Provide the customizable process panels displayed in the table format.
5. Thoroughly design all the process panels with no data entry size limitations.
6. Provide the on−line edit, browse, view, load, and unload functions for the VSAM files.
7. Provide the flexible and multiple methods for each single command function.
8. Allow the users to select all the previously executed commands for re−execution.
9. Allow the users to re−access all the previously edited files in a RECALL command panel.
10. Provide various simple commands for the switching between process panels.
11. Provide various markers to re−display the same screen of the process panels or edit panels.
12. Simplify several long and tedious TSO and ISPF command procedures.
13. Capture all the TSO on−line HELP tutorial text code lines in a panel format.
14. Simplify the XMIT command on both of the cataloged and uncataloged files.
15. Provide the simplified methods for the file copy, file move, and file comparison functions.
16. Provide the 'String Search' and 'Massive Change' functions on both sequential and PDS files.
17. Provide the multiple files edit, browse, and view functions in the ring format.
18. Allow the users to use their own User−Exit TSO/ISPF commands and Edit Macros.
19. Provide the Input Data and Balance Code automatic insertion function in edit files.
20. Provide the Comment Boxes and Lines automatic insertion function in edit files.
21. Provide the column insert, delete, and sort utility functions in edit files.
22. Provide many new invented and extremely powerful utility command functions.
23. Provide the Project Work Director function for accessing other member's profile.
24. Provide the File−AID tool package interface function for processing the VSAM files.
25. Provide the REXX source code formatter that is new to the MVS system.
26. Provide the REXXCHK function for a fast REXX program syntax error checking.
27. Provide the REXXREF function for generating the REXX source code cross reference.
28. Provide the DATE calculation utility function that is Year 2000 compatible.
29. Provide the detail on−line tutorial for all the essential command functions.

Each of the items described in the above Hightlight Summary will be discussed in detail later in this document.

1.1 The purpose of this tool package

Under the current existing ISPF working environment, if the users need to browse or edit a file, they may use the ISPF option 1, i.e. the Browse function, or use the ISPF option 2, i.e. the Edit function, or type a 'B'

(Browse) or an 'E' (Edit) command on the ISPF option 3.4 data set list panel, or type a 'B' or 'E' command on the ISPF option 3.1 library utility panel, or use the ISPF option 11, i.e. the ISPF Workplace function.

Although the users have so many choices to use the tools on the ISPF system, yet except for the newest ISPF option 11 function the users always need to remember the data set names they need to type in the ISPF control panels. Sometimes they need to spend a lot of extra time or effort to search for the data set names if they forgot them. Sometimes it is really a quite devastating situation if a user constantly types in the same data set name on the ISPF panel incorrectly and needs to fix it, and sometimes it is like attempting to open a combination lock without knowing the numbers. The users feel very frustrated with the slow methodology provided by the existing ISPF tool.

Some people created a VM file with their everyday used MVS data set name list and apply the Cut and Paste method between the MVS and VM file screens trying to avoid typing the very long and tedious data set names on the ISPF panels. However, that is really a very cumbersome and awkward way to solve the problems.

Some software groups developed their own ISPF tools to replace the ISPF option 3.4 data set list panel. On their locally developed ISPF panel it contains about 10 or 12 Dsname Level code entries for the users to choose among their most frequently used data set names. Therefore, whenever they select a new data set name they don't need to erase the previously entered data. This sounds like a very ingenious method. However, due to the total available Dsname Level code entries on the panel is too limited, this is still not the best solution.

The ISPF option 11, i.e. the ISPF Workplace function, allows the users to create their own 'Personal Data Set List' panels and define maximum up to 30 data set names on each panel, which is a little bit better than the local tools. However, since this function has such a panel size constraint problem, it is still not a perfect solution.

The purpose of this tool package is to provide various kinds of much easier and much superior solutions that can solve the inefficiency and inflexibility problems of the current ISPF system. This tool package has provided the customizable and virtually unlimited size panels which can assist the users to repeatedly work on the most frequently edited files very easily, and it allows the users to use their creativity to build their own TSO/ISPF commands and Edit Macros with the interface of the process panels provided by this tool package.

Although many functions provided by this MVS Project Work Manager tool package do not exist in the current ISPF system, yet the purpose of this tool package is not intended to completely replace the existing ISPF functions. As a matter of fact, when you use this tool, you may still be able to access the ISPF option 11, i.e. the ISPF Workplace panel, the ISPF option 3.4 data set list panel, and many other ISPF panels very easily. Therefore, there is no restrictions for the users to use both of this tool and the original ISPF tool functions at the same time.

Note: The ISPF option 11 is a very new option that has just been released by the ISPF Development Group recently in this year of 1998. Many MVS Software Development groups don't even know the existence of this new function yet. Some MVS Tools groups have already modified the ISPF primary menu many years ago and added an ISPF option 11 for other functions, such as the 'Book Manager' function. Those groups are urged to change the ISPF option 11 back to the 'ISPF Object/Action Workplace' function as soon as possible.

1.2 The methodology provided by this tool package

The following is the descriptions of the outlines of the newest methodology provided by this tool package:

1. The users can define as many projects or work items in a database file as they want. A process panel, i.e. the first process panel named the PANEL1 panel, has been provided for the users to enter the data to create the new projects or work items. The users can specify as many data set names to be associated with each project or work item as they like. Other than the data set names, the users can

also specify the TSO/ISPF commands, such as the batch job submit command, that are related to the same project or work item in the database file.

2. The project and work item names can be displayed on the second process panel named the PANEL2 panel, i.e. the main process panel of this tool package, for the users to select.

3. After a project is selected on the panel, all the data set names or the TSO/ISPF commands associated with the selected project can be displayed on another process panel, i.e. the third process panel named the PANEL3 panel, for the users to select.

4. Each User defined data set name and command name can be displayed on a panel with an unique identifier. This unique Identifier, i.e. the selection code, can be used as an index for the users to select a data set for edit, browse, or view, or select a command for execution.

   The selection code is in the 'a.b' format, where the code 'a' is the index of the project or work item, and the code 'b' is the index of the data set or the TSO/ISPF command. For example, the selection code '3.2' can represent the second data set of the third project defined by the user.

   This data set name selection code scheme is very similar to the ISPF option selection code, such as the most frequently used ISPF option code 3.4. However, the selection code provided by this tool is the User defined code and it can be changed very dynamically.

   If a new project or data set is added to the database or an old project or data set is deleted from the database and when the Fixlist Mode is turned Off, then the selection code scheme will be updated automatically or updated manually by the users when a 'RESET' or '/R' command is issued. More information about the Fixlist Mode will be described later in this document.

5. The fourth process panel named PANEL4 panel is a PDS Member List panel. This process panel has been provided for the users to either edit, browse, or view the member file of a PDS which is selected from the third process panel.

   There are two types of the PANEL4 panels provided by this tool package. One is in the original PDS Member List form, i.e. the long regular form. The other is in the short PDS Member List form. The PDS member names displayed on the short form panel are in a matrix format which condenses the entire PDS member list into only few panel screens. Each panel screen is called a screen page. The users may use a 'SW' (Swap) command to toggle between the displays of these two types of PANEL4 panels.

6. The users have the capability to enter a very simple command to switch from one project or work item panel to another panel.

   A Cross Reference process panel, i.e. the fifth process panel named PANEL5 panel, which contains a list of all the 'a.b' type of selection codes and their associated data set names or TSO/ISPF command names, has been provided. This panel not only can be used by the users to switch from one project or work item to another, but also can capture all the edit, browse, and view commands that were executed on the fourth process panel. The users can retrieve those edit, browse, or view commands and repeatedly work on the same edited, browsed, or viewed files on this process panel.

7. This tool package has provided the 'last used' project markers, 'last used' data set markers, 'last used' PDS member markers, and the edited file bookmark functions which allow the users to repeatedly work on the same file very easily.

   Note: The 'last used' project means the project or work item was most recently selected on the PANEL2 panel, the 'last used' data set means the data set was most recently selected on the PANEL3

panel, and the 'last used' member means the member file was most recently edited, browsed, or viewed on the PANEL4 panel.

This tool not only allows the users to repeatedly work on the most recently used file very easily, but also allows the users to display a specially designed panel to access all of the files that were previously edited in the past many days or many months. This is the so−called 'RECALL' command function. It is a very important function provided by this tool package. More detail information about the 'RECALL' command will be described later in this document.

8. This tool has provided a very good built−in new feature that can help you to avoid some mistakes. For example, when you select a file on the data set selection panel or on the ISPF option 3.4 data set listing panel, if you attempt to edit, view, or browse an unauthorized accessed data set by mistake, then the system will pop up a very irritating message to warn you that the owner of the data set has already been informed because you were trying to steal some information even though that was not your intention. If you enter a 'PWOPT' command to display a Miscellaneous Option Selection panel and set the 'The dataset access authority check' option on, then by using the 'E', 'V', or 'B' command on this tool panel, or by using the 'ED', 'VW', or 'BRW' command on the ISPF option 3.4 data set listing panel you will not fall into such bad traps again. When you use the 'E' or 'ED' command to edit a 'R/O' (Read Only) type of data set, this tool will automatically change the Edit command function to the View command function so that you cannot commit the mistake to update the unauthorized modified file.

9. This tool package has provided three very useful Edit Macro commands, i.e. the AUTOINS, CMTBOX, and CMTLINE commands. These commands can automatically insert the Balance Code, the comment box, and the remark comment line in the edited file running in the 'Automatic Insertion mode'. It can assist the users to write very neat program code in a very fast speed.

   Other than these three very useful Edit Macro commands, this tool has also provided the COLSHFT, COLUTIL, SORTUTL, and NODUP Edit macro commands for the enhancement of the ISPF Editor code text column shift, and general column utility functions and the code sorting utility functions, which are all very useful functions.

10. This tool package has provided the 'XSPLIT' Edit Macro function, which has been assigned to the PF2 key, for the edit text data split and join functions. This 'XSPLIT' edit macro performs just like the regular ISPF 'SPLIT' command to split the ISPF screen if the cursor remains on the edit command line or prefix line command area when the PF2 key is pressed. However, if the cursor is placed in the edit file area when the PF2 key is pressed, then this PF key will split or join the text data in the edit file depending up where the location the cursor is placed.

11. This tool package has provided the 'XCUT' and 'XPASTE' Edit Macro commands, which have been simplified to the 'G' (Get) and 'P' (Put) command names, for the ISPF Editor code text Cut and Paste functions. It has also provided two similar functions, which are the 'GETVAR' and 'PUTVAR' Edit Macro commands, that allow the users to copy and paste the edit screen text data and they can help the users to save a lot of time in developing the program code.

    The 'GETVAR' and 'PUTVAR' edit commands have been assigned to the PF9 key, i.e. the 'XSWAP' command key. If the ISPF screen has already been split, then the 'XSWAP' command performs just like the regular ISPF 'SWAP' command if the cursor remains on the edit command line or prefix line command area when the PF9 key is pressed. If the cursor is placed in the edit file area when the PF9 key is pressed, then this PF key will Copy or paste the text data in the edit file depending up where the location the cursor is placed, and it works just like the extended Cut/Paste command functions.

12. This tool package has provided an 'Edit Ring' function on the PANEL4 List panel and the RECALL command panel. It is a simulation of the 'VM XEDIT Edit Ring' function, which allows the users to

edit several PDS member files on the PANEL4 panel or edit several data sets on the RECALL command panel simultaneously in an Edit Ring and apply the 'Cut' and 'Paste' commands to copy or move the code among several edited files very easily.

The same method has also applied to the multiple browsed files and viewed files in the 'Browse Ring' format and the 'View Ring' format.

13. In addition to the five process panels for the Project Work Manager function, this tool package has also provided a special panel for the Project Work Director function which allows the users to access the data sets or execute the TSO/ISPF commands organized by other members in the team. This is a very useful feature which can help the members of the entire Software Development organization to work together synchronously. The Project Work Director panel can be accessed when a 'DIR' command is entered on any of the five process panels in the Project Work Manager tool session.

14. This tool package has provided the 'LISTA' and 'LISTC' commands to allow the users to display all the allocated data sets and cataloged data sets of the TSO logon session on a panel so that the users can manipulate those data sets very easily.

15. This tool package has provided an 'Extended String Search' utility which allows the users to select multiple input data sets on the PANEL3, LISTA, LISTC, and RECALL command panels to perform the string search function.

    On the 'Extended Search' viewed output listing, the users may apply the PF4 key, i.e. the 'ED' command function key, to edit many PDS members, which contain the searched data strings, one by one in an Edit Ring to perform the 'Massive Change' function.

    The 'ED' command has been extensively modified and it can be applied to many different utility functions provided by this tool package. More detail information about the usages of each function that uses the PF4 key (i.e. the 'ED' command function) can be found in this document.

    Note: As the 'ED' is a very common edit macro name, the 'EDT' command has been used to perform the same 'ED' command function.

16. This tool package has provided an "User−Exit" function that allows the users to define their own TSO/ISPF commands and Edit Macro commands to be used with the interface of the process panels provided by this tool package.

    For example, the two User−Exit commands named 'FL' and 'FF' provided by this tool package can be used to display the ISPF option 3.4 data set list panel for the users to use all of the original ISPF commands on the MVS files within a single Project Work Manager tool session. The 'FF' command can also be used to display an ISPF option 3.4 Front−end Interface panel for the users to define many Dsname Level codes in a database file.

17. This tool package has provided several ISPF window pnel facilities, i.e. the 'MSGBOX', 'MENUBOX', 'ITEMBOX', and 'HELPBOX', which can be used to display the ISPF window panels and allow the users to display their own customized messages, menus, item selection list, and tutorial guide.

18. This tool package has provided the 'WORDPRF' utility that can be used for the word spelling check and words proofreading functions of the MVS SCRIPT files.

19. This tool package has provided the 'CODEDIFF' utility that can be used for comparing two program source files and generate the very neat side−by−side result output listing file.

20. This tool package has provided several different types of file transmit command functions, i.e. the 'SENDFILE', 'PRTPDS', 'SENDPDS', 'SENDSEQ', and 'SFX2MVS' command functions. Those are all very User−friendly and very powerful functions which are all very easy to use. They can help the users to greatly improve their work performance.

21. This tool package has provided a 'UPDPDS' command function that can be used to compare a Source PDS with a Target PDS and create a Incremental Delta PDS file that contains the members of the Source PDS which are different from the members in the Target PDS. This Delta PDS file can be used for updating the Target PDS, and it is a very good tool for the MVS system programmers and administrators.

22. The current ISPF tool cannot handle the on−line edit and browse functions of the VSAM files. However, there is a Vendor Software Company provided tool package named File−AID can handle such functions very well. This Project Work Manager tool package has provided a 'FASETUP' command for the users to create a FILEAID profile to build up an interface between the File−AID function with this tool function. The users can use this function to edit or browse the VSAM files on−line without any difficulties.

    If the File−AID facility is not installed on the MVS system, the users may still use this tool package to perform several fundamental on−line functions, such as the VSAM file edit, browse, view, load, and unload functions.

23. This tool package has also provided various kinds of new utility program functions, such as the REXX source code formatter function, the Arithmetic and Date/Time calculation function, the monthly calendar and the Things−To−Do function, and the Notebook function, etc. which are all extremely useful functions.

This tool is in a state of art because once you have done the setup of the front−end database files you don't need to memorize those long and tedious data set names that you have to deal with in your everyday works any more. All you need to do is to type some selection codes, click the mouse button and press the Enter key on the panel, or enter some very simple commands to select the files.

Suppose you cannot remember the simple command names that this tool has provided, then you may enter a 'CMD' command on the PANEL2, PANEL3, and PANEL4 process panels to display a Command List panel, which is very much like a pull−down panel, for you to perform various kinds of command functions.

Most of the TSO/ISPF commands and Edit Macro functions provided by this tool are extremely useful. They have already been closely tied−up with this tool package and cannot be used as stand−alone tools or attached to any other tool packages. Please note that to detach the functions from the tool libraries of this tool package and use them independently without the interface with the main routine is not recommended because it won't work. In order to use those tool functions properly, you must establish the MVS Project Work Manager tool session first. The purpose of this restriction is to ensure the integrity of functions provided by this tool package and guarantee the users can take the full advantages provided by this tool package.

This restriction, however, does not mean that you will lose the opportunity to use the original functions of the traditional ISPF system. By using this tool, you may still be able to apply all the functions provided by this tool package to the original ISPF system.

In the last section of this document it contains the descriptions of the tool procedures that I frequently used. It can be served as a good reference for you to use this tool.

1.3 The installation of this tool package

To use this Project Work Manager tool package, initially you need to create the six primary library files on your TSO account. To create these six library files, you may receive them either from the VMTOOLS libraries or from the MVSTOOLS libraries, or receive them from your friends by using a 'RECEIVE' command and install them on your TSO account.

1. If the received files are already in the PDS format, then the installation is done.

2. If the received files are in the 80 bytes fixed block sequential file format, which was created by using the 'SENDPDS' command function, then you need to use the 'RECEIVE INDA(/)' command on the ISPF option 3.4 data set list panel to restore them back to the PDS library files.

3. If the received files are in the 84 bytes fixed block sequential file format or in a variable record length sequential file format, which was created by the 'SENDPDS' command function with the 'Line Splitter' code on each code line and transmitted to you through Internet or uploaded from CD or IOMEGA Zip 100 diskettes, then you need to use a XRSTOR (i.e. The XMIT File Restore) batch job process or use a 'XRESTORE' command to convert them to the 80 bytes fixed block format, and then use the 'RECEIVE INDA(/)' command on the ISPF option 3.4 data set list panel to restore them back to the PDS library files.

Note: A sample XRSTOR batch job JCL file and a sample XRESTORE program have already been provided in the DOCUMENT library of this tool package for your reference.

Then on the 'COMMAND' field of your TSO logon panel you need to either specify the following EXEC command:

```
EXEC 'USERID.@PROJWRK.CEXEC(@@INIT)'
```

or specify the following EXEC command:

```
EXEC 'TOOLKIT.@PROJWRK.CEXEC(@@INIT)'
```

which depends upon whether this MVS Project Work Manager tool package is installed on your TSO account or installed on a tools library account such as the 'TOOLKIT' account. The setup of the TSO logon panel is needed to be done only once. You may take it off from the TSO logon panel when you are not using this tool package.

Note: If your tools administrator has already made the 'TOOLKIT.@PROJWRK.CEXEC' library to be accessible by everybody in the Software Development organization, then you don't need to specify anything on your TSO logon panel and you don't need to insatll this tool package on your TSO account. To accomplish this, the tools administrator must copy the 'PROJ' file from the 'TOOLKIT.@PROJWRK.CEXEC' library to a logon CLIST library that will be used by everybody in the organization.

Note: When the tools administrator initially started to install this tool package, he or she needs to check the two member files, i.e. the '@@INIT' and 'PROJ' files, in the 'TOOLKIT.@PROJWRK.CEXEC' library and make sure these two files have been setup properly. He or she may also use the sample 'ISR@PRMX' file in the 'TOOLKIT.@PROJWRK.PANELS' library as a reference to create a similar ISPF primary main panel file named 'ISR@PRIM' for everybody in the organization to use.

Note: If you don't have a tools administrator to setup this tool for you, and you don't want to use the '@@INIT' file in the CEXEC library on your logon panel, then the alternative method is to copy the 'PROJ' file from the 'TOOLKIT.@PROJWRK.CEXEC' library by yourself to your own logon CLIST library.

After the setup of the TSO logon panel is completed, then simply enter a 'TSO PROJ' command on any of the ISPF panel command line to start to use this tool.

As you can see that this tool package has provided a special method which can make all the MVS library files transportable to any MVS Software Companies in the world through various kinds of systems such as VM, Lotus Notes, or Internet. More detail information about transporting the MVS tool packages can be found in this document.

In the 'USERID.@PROJWRK.DOCUMENT' library provided by this tool package, there is a 'HTML' file available for you to view this document on−line by using a Netscape or Microsoft Browser. Please see the '@README' file in this DOCUMENT library for more information about this tool package.

Note: There are several very important things need to be noticed when you try to install this tool package.

1. If you specify the following EXEC command on your TSO logon panel:

   ```
   EXEC 'USERID.@PROJWRK.CEXEC(@@INIT)'
   ```

   where the 'USERID' is your TSO logon Userid, then please don't try to simplify it to the EXEC command as the following because it won't work:

   ```
   EXEC @PROJWRK.CEXEC(@@INIT)
   ```

   The TSO system will complain that the 'USERID.@PROJWRK.CEXEC.CLIST(@@INIT)' file cannot be found because the TSO system assumes that it is a CLIST data set you are trying to execute when the data set is not enclosed in single quotes.

2. If you specify the following EXEC command on your TSO logon panel, for example:

   ```
   EXEC 'TOOLKIT.@PROJWRK.CEXEC(@@INIT)'
   ```

   and if the TSO system issues an error message such as the following:

   ```
   Phase I processing of CLIST or REXX EXEC ended abnormally +
   ```

   Then you need to check whether or not your TSO account is authorized to access the tool library system, such as the 'TOOLKIT.@PROJWRK.CEXEC' library. If you cannot obtain the R/W or R/O access authorization of the tools from your system tools libraries, then you may consider installing the six primary tool libraries of this tool package on your own TSO account instead.

3. When the TSO system executes the 'USERID.@PROJWRK.CEXEC(@@INIT)' command file and if it issues an error message such as the following:

   ```
   The run time processor EAGRTPRC is not available +
   ```

   Then it means that the REXX run−time library has not been installed on your MVS system. In this

case, you cannot use this tool package. The REXX run−time library is usually named 'SYS1.REXX.SEAGLPA'. It is a load module library which is usually installed on the MVS system by the MVS system programmer.

2.0 The file structure of this PROJWRK tool package

The following few sections contain the brief descriptions of the structure of this tool package. The material in this part was basically extracted from the Reference Guide. If you need to know more about the structure of this tool, please read the Reference Guide for details.

2.1 The descriptions of the tools library files

This tool package contains six primary library files, which are:

- 'USERID.@PROJWRK.CEXEC',
- 'USERID.@PROJWRK.PANELS',
- 'USERID.@PROJWRK.TABLE',
- 'USERID.@PROJWRK.SKELS',
- 'USERID.@PROJWRK.DOCUMENT', and
- 'USERID.@PROJWRK.LOAD'.

where

- the CEXEC library contains the executable process control modules and utility modules, which are the compiled version of the REXX programs;
- the PANELS library contains the process panel source code;
- the TABLE library contains a PF key definition file and the command list files;
- the SKELS library contains the batch job JCL skeleton files;
- the DOCUMENT library contains the tutorial guide of this tool package and the source code of the XRSTOR batch job JCL file; and
- the LOAD library contains several load modules that are used by some commands provided by this tool.

After this tool is invoked at the first time, it will create two library files on your TSO account, which are:

- 'USERID.@PROJWRK.LIST',
- 'USERID.@PROJWRK.XREF', and
- 'USERID.@PROJWRK.DATA'.

where

- the LIST library is a database file that contains the specifications of the projects or work items;
- the XREF library contains various kinds of database files for the Cross Reference function, the Fixlist function, the User−Exit function, and the file transmit function, etc.; and

- the DATA library is a database file that contains the tersed LIST file for backup purpose. It can also be used for storing the tersed data of the TODOLIST and NOTELIST command functions.

Therefore, totally there are eight library files of this tool package − six are the primary tool library files and two are the tool generated library files.

Note: If this tool package is not installed on your own TSO account, and if you don't have the Write Access Authority to update the CEXEC, SKELS, PANELS, and the LOAD library files of this tool package, then you may create any of the following four User−Exit files on your own TSO account:

- 'USERID.@USREXIT.EXEC',
- 'USERID.@USREXIT.SKELS',
- 'USERID.@USREXIT.PANELS', and
- 'USERID.@USREXIT.LOAD'

In your 'USERID.@PROJWRK.XREF(PROFILE)' file, suppose you have changed the control code for the 'USEREXIT' option from 'NO' to 'YES', then during the initial bring−up process of this tool, when this tool detects the existence of any of these four user−defined library files it will automatically concatenate them to the tools library files of this tool package. Please see the Appendix A section for the detail information about how to specify the control code in the user−defined 'PROFILE' file in your personal XREF library.

2.2 The descriptions of the process panels

This tool package contains five process panels, which are:

1. The PANEL1 panel − **'The Project or Work Item Creation Panel'**

   This panel can be used to create a Project List member. In the Project List member file you may specify various kinds of valid data set names and/or the TSO/ISPF command names.

2. The PANEL2 panel − **'The Project or Work Item Selection Panel'**

   This is main process panel of this tool package. This panel allows you to select a project or work item. The PANEL2 panel is also the only process panel on which you may enter a 'S newproj' command to display the PANEL1 panel for creating your new project or work item.

3. The PANEL3 panel − **'The Data Set or Command Selection Panel'**

   This is a data set name list panel. It can be used to select a Partitioned, Sequential, GDG base file, or VSAM file for the edit, browse, or view function. On this panel you may also select the valid TSO/ISPF command for execution.

4. The PANEL4 panel − **'The PDS Member List Panel'**

   This is a PDS member list panel. If a PDS is selected on the PANEL3 panel, then all the PDS members will be displayed on this panel for the edit, browse, or view function.

5. The PANEL5 panel − **'The Xref List and Retrieved Commands Panel'**

   This panel contains a Cross Reference list of all the data sets of the projects or work items that you

have defined. It also contains a list of previously executed Edit or Browse commands that you can retrieve them and execute them repeatedly.

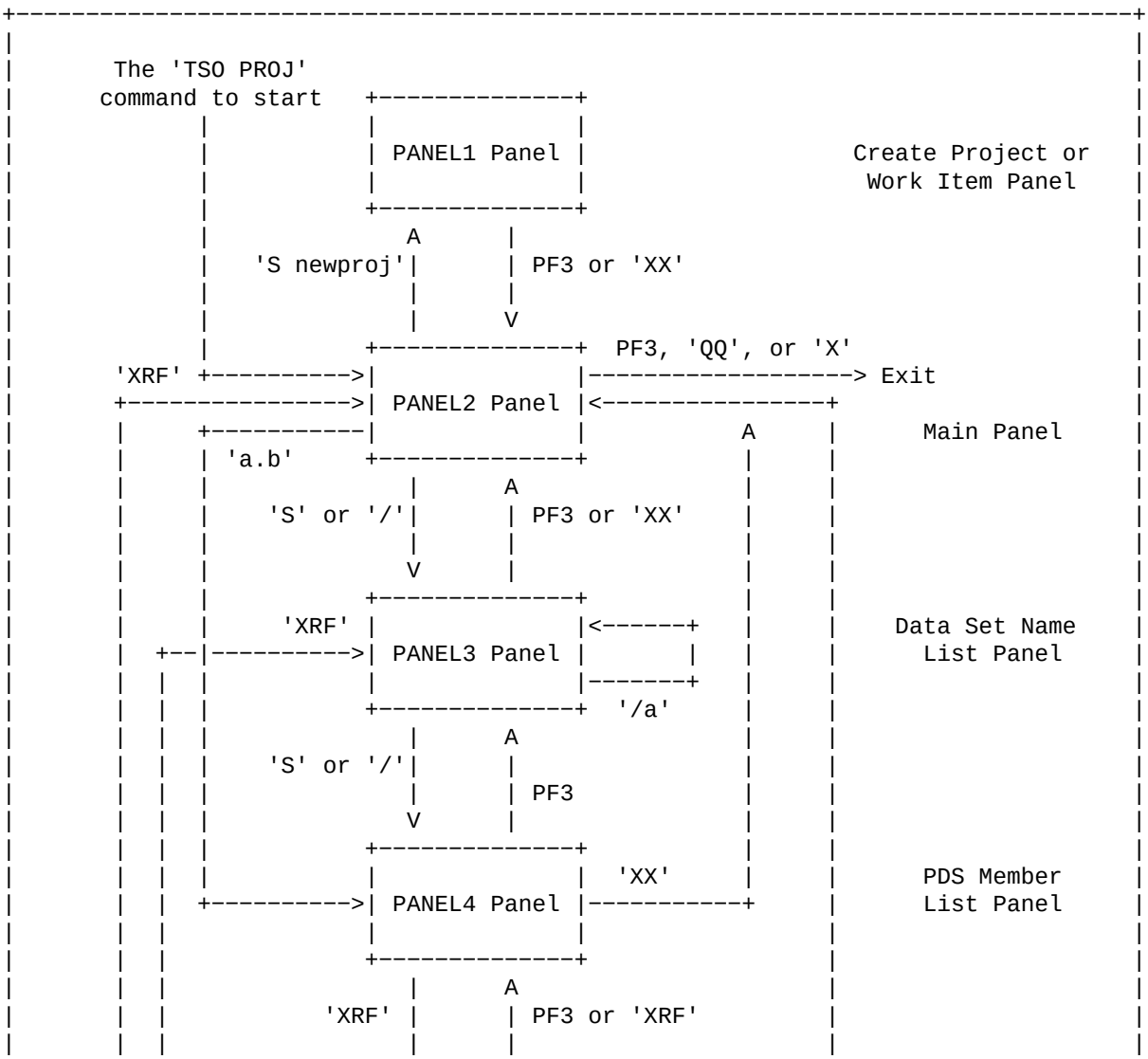
The above described five process panels can constitute a Project Work Manager tool session. In addition to these five process panels, this tool package has provided two other very important process panels, which are:

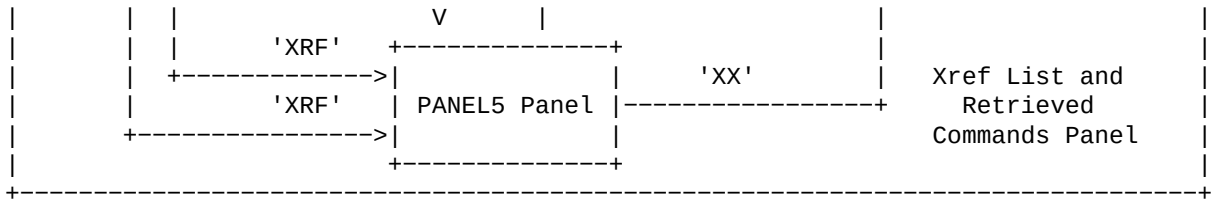
1. The PROJDIR panel, i.e. the **Project Work Director** function panel. This panel can be used for accessing the MVS data sets organized by other person's Project List file in another Project Work Manager tool session. It is a control panel which allows you to access the Project Work Manager tool sessions of different TSO accounts very easily.

2. The RECALL command panel, which can be used to display all the sequential files and PDS member files that were previously edited on the PANEL3 or PANEL4 panel.

2.3 The graphic representation of the process panels structure

The following is the graphic representation of the relationship among these five process panels in a single Project Work Manager tool session:




V
'XRF'
'XRF'
PANEL5 Panel
'XX'
Xref List and
Retrieved
Commands Panel

3.0 The methods of how to use this PROJWRK tool

The following is a list of the descriptions of the basic functions that can assist you to learn how to use this PROJWRK tool. The process steps of each function in this document are all described very briefly. To obtain more detail information of each function, please refer to the Reference Guide. The Reference Guide contains the information of many advanced and extended ISPF functions provided by this tool package which are not covered in this document. It is also highly recommended that you may read the Highlights document of this tool package for a quick overview of the functions of this tool package also.

Before reading this document, please be sure that this tool package has been properly installed on your MVS system. The detail information of how to install this tool can be found in either the Reference Guide or the Highlights document of this tool package.

Note: Both the material of this document and the Highlights document were originally extracted from the same resource, i.e. the Reference Guide document. Therefore, when you read these two documents you will see the redundancy of the descriptions of some functions. However, for those repetitively described functions it means that those functions are more important than the others.

This PROJWRK tool package is a very large MVS tool and it contains many new functions and new concepts that you may not have seen from any other MVS tools before. Since most of the functions provided by this tool package are basically the simplifications of the original ISPF tools, therefore, if you are already familiar with the existing ISPF system, then after you finish reading this document you will probably find that this tool package is actually very easy−to−learn and very easy−to−use.

3.1 How to invoke this PROJWRK tool

If you have already properly setup the logon procedure to link to this tool on your MVS system, then you may simply enter a **"TSO PROJ"** command on the ISPF Primary Menu command line to invoke this tool.

When this tool is invoked, the PANEL2 main process panel will be displayed.

Note:

1. If your ISPF Primary Menu contains a 'D' option for the **"Alt Dialog"** function, then you may fill in that option with a "CMD(%PROJ)" command, and then you can choose the 'D' option instead of using the 'TSO PROJ' command to invoke this tool. However, if your ISPF Primary Menu does not have such an alternative Dialog option, then you may make a copy of the ISPF Primary Menu in your personally defined ISPF panels library on your own TSO account and add an new ISPF option, such as the ISPF option 12, for the PROJWRK function. This method is a little bit complicated. Please see the Reference Guide for more detail information about this method.

2. This tool contains a lock for preventing it from being redundantly invoked. Thus, if the Project Work Manager is already in process, it is not allowed to issue another 'TSO PROJ' command again within the Project Work Manager tool session. Otherwise, a **'Projwrk tool locked'** error message will be prompted.

3. If the Project Work Manager tool session is terminated due to an internal program error occurs, then the tool lock will also be locked. When you re−enter a 'TSO PROJ' command again, the same message of **'Projwrk tool locked'** will be shown. In this case, to re−open the tool lock you may enter a **'TSO PROJ UNLOCK'** command.

3.2 How to create a new project or work item on the PANEL1 panel

1. On the PANEL2 panel when you enter a "S newproj" command, where 'newproj' can be any new project or work item you choose, then a PANEL1 panel will be displayed.

   Note: Other command form of the "S" (Select) code in the "S newproj" command is the "N newproj" or "CR newproj" command, which stands for the "New newproj" and "CReate newproj".

2. On the PANEL1 panel, you may type "This is test" in the function area and type "USERID.*" on the data set name field as an example, where 'USERID' is your TSO logon Userid.

3. The data set name field may contain a Dsname level code with an "*", or a valid fully qualified data set name without specifying the bounded single quotes.

4. After pressing the Enter key, a new project or work item will be created.

Note:

1. On the PANEL1 panel you may specify the valid PDS, sequential, GDG base, and VSAM data set names. The data set name must be fully qualified and not be bounded by single quotes. For the uncataloged data set, you must specify the associated Volume Serial number. In the Valid TSO/ISPF Command field, you may also enter any valid TSO/ISPF command that starts with the 'TSO', 'EXEC', 'PGM', 'CMD', 'PANEL', 'ISPEXEC', 'EDIT', 'BROWSE', or 'VIEW' keyword code.

2. There are two other methods, such as the 'ADD' command method and the 'PDS Member Copy' method, can also be used to add new data set names in an existing project or work item, or create a new project or work item. These two methods will be described later in this document.

3.3 How to select a project on the PANEL2 panel

1. Enter a selection code (which can be entered on either the PANEL2 panel command line or on the PANEL2 panel selection code line), or

2. Type a 'S' or '/' next to a project, or

3. Position the cursor at any column on a project or work item code line and press the Enter key

can all be used on the PANEL2 panel to select a project or work item.

If a project or work item has already been flagged with a "Last Used" marker, i.e. the "−" marker, then simply press the Enter key while the cursor is still on the PANEL2 panel command line to select the same project or work item.

Note: On the PANEL2 panel command line, you may use a command, such as the "E dsname" command, to edit a cataloged data set named "dsname" that is not defined in any project or work item. Thus, the PANEL2 panel command line can be used as a temporary replacement of the ISPF option 3.4 data set list utility panel for a cataloged data set. For example, if you enter a command such as "E logon.clist", then the data set 'USERID.LOGON.CLIST', where 'USERID' is your TSO logon Userid, will be edited.

3.4 How to select a data set on the PANEL3 panel

1. Enter a selection code (which can be entered on either the PANEL3 panel command line or on the PANEL3 panel selection code line), or

2. Type a 'S', '/', or 'E' command code next to a data set to edit the data set, type a 'B' code to browse the data set, or type a 'V' code to view the data set, or

3. Position the cursor at any column on a data set code line and press the Enter key

can all be used on the PANEL3 panel to select a data set.

If a data set has already been flagged with a "Last Used" marker, i.e. the "−" marker, then simply press the Enter key while the cursor is still on the PANEL3 panel command line to select the same data set.

Note:

1. On the PANEL3 panel command line, you may enter an 'E' command to edit the Project List member file of the selected project or work item to add or delete the data set names or TSO/ISPF command names. After the Project List member edit session is finished, the PANEL3 panel will be automatically refreshed.

2. In the edited Project List member file, instead of deleting the data set names or TSO/ISPF command names, it is recommended that you may position the cursor on the code line and press the PF4 key to comment it out. By pressing the PF4 key again on the same code line the inserted "/*" code at the leftmost column on the code line can be dropped. Thus, the PF4 key can be used to toggle between the '/*' code to be inserted into and dropped from the leftmost column of the code line in the Project List member file.

3. You may use the same procedure to enter a selection code, type a 'S' or '/' code, or use the cursor and press Enter key to select a TSO/ISPF command on the PANEL3 panel.

4. On the PANEL3 panel data set name area if you place the cursor in the middle of a data set name and press the PF4 key, i.e. the RECALL command key, or the Enter key, then the data set name will be expanded using the code to the left of the cursor as the search pattern, and a selection panel of the expanded data set name list will be displayed.

   On any process panel you may enter a 'PWOPT' or an 'OPT' command to display an Option Selection panel and alter the default value 'OFF' in the 'Dataset Access Authority Check' field to 'ON'. Then whenever you type an 'E' command next to a file which you are not authorized to update, then this tool will automatically convert the edit command function to the view command function unless an 'EDX' command is used. However, to use an 'EDX' command to edit an unauthorized file is not recommended.

Note: The ISPF Workplace Personal Data Set List function is very similar to the PANEL1, PANEL2, and PANEL3 panel functions. The problem of the Personal Data Set List function is that it has the panel size limitation. Thus, if you have a project or work item that contains more than 30 data set names, then you have to split them into several Personal Data Set List panels and it is very hard for the users to remember which data set name of the same project is specified on which panel. Furthermore, you are not allowed to specify any frequently used TSO/ISPF commands on those panels. The PROJWRK process panels, on the other hand, do not have such kinds of restrictions. However, if the total amount of the data entries on a PANEL3 panel is too large, it will slow down the process.

On the Option Selection panel you may alter the default value 0 in the 'Maximum PANEL3 Migrate entries' field to a larger value, say the value of 50. Whenever a PANEL3 panel contains less than and equal to 50 data entries the Volume Serial field of the migrated data sets will be shown either a code of 'MIGRAT1' (Migrated to a DASD) or 'MIGRAT2' (Migrated to a Tape). If the data set is not cataloged, then a '−−' flag will be indicated. However, a large number in the 'Maximum PANEL3 Migrate entries' field will slow down the PANEL3 panel display process significantly. Thus, to keep this option with a default value 0 is highly recommended.

3.5 How to process the PDS members on the PANEL4 panel

1. On the PANEL4 panel, you may move the cursor down along the member command line area and position it next to a selected member, and then press the Enter key to edit the member. If an 'E', 'S', or '/' code is typed in next to the selected member before the Enter key is pressed, then the member will also be edited. The 'E', 'S' or '/' command can be omitted.

2. If you want to browse or view the member, then you must type in a 'B' or 'V' code on the member command line next to the selected member and press the Enter key. The 'B' and 'V' command code cannot be omitted.

3. If a member has already been flagged with a "Last Used" marker, i.e. the "−" marker, while the cursor is still on the PANEL4 panel command line area, then simply enter an 'E' command on the PANEL4 panel command line and press the Enter key to edit this most recently edit member file. This 'E' command can be omitted.

   You may enter a 'B' or 'V' command and press the Enter key while the cursor is still on the PANEL4 panel command line to browse or view the most recently browsed or viewed member which has been flagged with a "−" marker.

   Note: Similar to the methods of choosing the 'Last Used' project on the PANEL2 panel and choosing the 'Last Used' data set on the PANEL3 panel, if you may just press the Enter key on either the long form or the short form PANEL4 panel without specifying any of the 'E', 'B', and 'V' commands, then this tool will redo the last edit, browse, or view command on the 'Last Used' PDS member. The short form PANEL4 panel will be discussed later in this document.

4. Enter a '/' command on the PANEL4 panel command line can locate the most recently used member file if the member list on the PANEL4 panel is very large and you don't see a "−" marker on the current PANEL4 panel screen.

5. After several PDS members are edited, browsed, or viewed, you may use the '\' or '\\' command on either the PANEL4 panel command line or member command line to backtrace the members from a PDS Backtrace buffer. Note that you may use a '\\\' command to reset the buffer to empty. The '\', '\\', and '\\\' commands also work in the edited or viewed PDS member files.

6. All the original ISPF member line commands, such as 'E' (Edit), 'B' (Browse), 'V' (View), 'C' (Copy), 'M' (Move), 'D' (Delete), etc. are all applicable to the PDS Member List on the PANEL4 panel.

7. On the PANEL4 panel you may also use a pair of 'DD' code to delete several member files at the same time. You may also type a command such as 'D4' or 'D 4' command, for example, to delete four members below the member cursor line. The 'D999' command can be used to delete the rest of members below the member cursor line where the command is typed in.

   Note: The 'DD' and 'D999' commands are very handy. However, there will be no warning message window panel popped up to warn you about it. Thus, when you apply these types of commands, you

have to be very cautious to prevent deleting some PDS members by mistakes.

8. You may type few 'E' code next to several member names on the PANEL4 panel to edit these selected member files at the same time.

9. This 'E' command method is different from the 'Edit Ring' method. On the PANEL4 panel command line you may enter a 'RING' command and use the 'S' code to select the PDS members to edit several member files at the same time in an Edit Ring format.

10. The 'SELECT' command is similar to the 'RING' command method, but there is a difference between these two commands. For example, when you enter a 'S PROJ*' command on the PANEL4 panel, the PDS members whose first four characters match the string 'PROJ' will be edited one by one. However, after the last selected member is edited, the first member will not be edited again. The 'RING PROJ*' command allows you not only edit each member whose name matches the string 'PROJ' in a Edit Ring format, but also allows you to press the PF4 key to go backward to edit the previously edited member files in the ring. Thus, the 'RING' command method is much more flexible.

11. If you type an 'E' code next to a PDS member file which you are not authorized to update, then this tool will automatically convert the edit command function to the view command function unless an 'EDX' command is used. However, to use an 'EDX' command to edit an unauthorized file is not recommended.

12. You may use either a 'S TEMP' or 'E TEMP' command to create a new PDS member named 'TEMP' on the PANEL4 panel.

3.6 How to use 'FF', 'FL', and few other commands on the PANEL3 panel

1. You may use the 'F data' command on the PANEL3 panel to find a data string in order to search for the data set names or command names.

2. Type the 'FF' command next to a data set name on the PANEL3 panel can display the ISPF option 3.4 data set list panel of the selected data set.

3. Next to a data set on the ISPF option 3.4 data set list panel and type an 'ADD' command can add the data set name in a selected Project List member file.

4. Instead of using 'ADD' command to add the data set names into the Project List member file, you may also edit the member file and type a partial data set name and then use the 'EXPAND' edit command to generate the full qualified data set name list. More detail information of this advanced method can be found in the 'How to use the EXPAND Edit Macro' section.

5. Next to a data set on the ISPF option 3.4 data set list panel you may use the 'CP' or 'MV' command to display ISPF option 3.3 panel and copy or move the selected data set to a new or an existing data set.

6. Use the 'FL' command next to a data set on the PANEL3 panel will display the ISPF option 3.4 data set list utility panel of the selected data set. You may try out both 'FF' and 'FL' command and you will easily find out the difference between these two commands.

7. The 'FF' and 'FL' commands can also be entered on the PANEL4 panel command line area or on a PDS member command line area.

8. The 'ADD' command can be applied to the ISPF option 3.4 data set list panel displayed by using the 'E dsn' command on the PANEL2 panel command line also, where 'dsn' is a data set name code that is not defined in any project.

9. Use the 'LIB' command next to a data set on the PANEL3 panel can display the ISPF option 3.1 panel.

10. Use the 'DS' command next to a data set on the PANEL3 panel can display the ISPF option 3.2 panel.

3.7 How to use 'Z' command to compress a PDS

1. You may type a 'Z' command next to a PDS on the PANEL3 panel to compress that partitioned data set.

2. You may use a 'Z' command and several '=' command code on the PANEL3 panel data set select code lines to compress several PDS files at the same time.

3. Use the 'Z' command on the PANEL4 panel command line can compress a PDS also.

4. In edit, you may enter a 'EDX' command to display a PANEL4 panel of the PDS and then use 'Z' command to compress it. Note that when you view a PDS member file, you may use the 'EDX' command to display the PDS and then use the 'Z' command to compress it also.

5. Similarly, in edit you may enter a 'RC' command to display the RECALL command panel and then use an 'X' command next to a PDS, which has a member name associated with it, to display the PDS and then you may use 'Z' command to compress the PDS.

6. In the above described cases no ISPF panel screen split is necessary when you compress the PDS files in edit. The 'EDX' and 'RC' commands are very much like the external communicator for edit and they allow you to get across the boundary and enter into the outside of the edit session.

7. The two PROJWRK created library files, i.e. the 'USERID.@PROJWRK.LIST' and 'USERID.@PROJWRK.XREF' files, are the PDS−E type of library files which do not need to be compressed at all.

3.8 How to use 'C' and 'M' commands on the PANEL3 panel

1. Use the 'C' command next to a data set on the PANEL3 can copy the selected file to a target file which can be either an existing file or a new file.

2. When the selected file on the PANEL3 panel is a PDS, then the target file to be selected or created must be also a PDS. If the select file is a sequential file, then the target file must be also a sequential file.

3. When the 'C' command is type next to a data set on the PANEL3 panel, then a window panel will be popped up. If the selected file will be copied to an existing file, then the ISPF option 3.3 utility function will be invoked.

4. You may type a 'C' command next to one data set and type a 'S' (Select) code next to the target file on the PANEL3 panel to perform the copy file function if both data set names can be found on the same PANEL3 panel screen.

5. Instead of using the 'S' to select the target file, you may press the PF11 key on the popped up window panel to display a 'SELPROJ' panel and select the target file from that panel.

6. If the target is an existing file but it cannot be found on the 'SELPROJ' panel, instead of pressing the PF11 key to search for it you may type the target file name partially on the panel ended with a '/' code and press the Enter key to expand the data set name and display the data set name list on a selection panel, which is named EXPDSNS panel. On that panel you may type one and only one 'S' code and press the PF3 key to select a target data set name.

7. Note that the '/' code can be inserted in the middle of the target data set name field because when this tool expands the data set name it only pick up the data to the left of the '/' code as the data name match search pattern.

8. If copy to a new file, then a popped up window panel with the new file structure will be displayed. You may alter the data on the window panel to create the either a larger or a smaller new file. After pressing Enter key the new file will be automatically created without using ISPF option 3.1 and 3.2 functions.

9. Use the 'M' command on the PANEL3 can move a PDS or sequential file which can be either an existing file or a new file. This method is similar to the 'C' command.

10. On the ISPF option 3.4 data set list panel, which is displayed by using the 'FF' command from the PANEL3 panel, you may use the 'CP' or 'MV' command to copy or move the data set.

3.9 How to handle the VSAM files on the PANEL3 panel

1. Next to the CLUSTER file (or a INDEX, or DATA file) on the PANEL3 panel you may type an 'E', 'B', or 'V' command to edit, browse, or view the file.

2. If a Vender Software Company developed File−AID tool package has been installed on your MVS system, and if you have properly setup the interface with the FILEAID tool package, then this tool will automatically switch to use the FILEAID tool package to perform the VSAM file edit or browse function. Note that the view function is not support by FILEAID.

3. If your MVS system does not have the FILEAID facility, then you may still use the VSAM file on−line edit, browse, and view functions provided by this tool package.

4. Next to the CLUSTER file (or a INDEX, or DATA file) on the PANEL3 panel you may type an 'C' command to copy the VSAM file to a new created sequential file. This is the so−called VSAM file unload function.

5. Next to the sequential file on the PANEL3 panel you may type an 'C' command to copy the sequential file to an existing VSAM file. This is the so−called VSAM file load function.

6. On the ISPF option 3.4 data set list panel, you may use the 'ED', 'BRW', or 'VW' command to edit, browse, or view the VSAM file. You may also use a 'CP' command next to a valid sequential file or a VSAM file to perform the VSAM file load or unload functions.

3.10 How to use 'C' and 'M' commands on the PANEL4 panel

1. You may type a 'C' command next to a member on the PANEL4 panel and enter the target member name at the "User Data" area on the same member name line if the target is a new member.

2. If the target is not a new member, then be sure to leave the "User Data" area blank and press the Enter key. Then a window will be popped up.

3. On the popped up window, you may fill in the target PDS data set name and target member name manually, or press PF11 key to display a 'SELPROJ' panel to select a target PDS data set from that panel.

4. If the target is an existing file but it cannot be found on the 'SELPROJ' panel, instead of pressing the PF11 key to search for it you may type the target file name partially on the panel ended with a '/' code and press the Enter key to expand the data set name and display the data set name list on a selection panel, which is named EXPDSNS panel. On that panel you may type one and only one 'S' code and press the PF3 key to select a target data set name.

5. Note that the '/' code can be inserted in the middle of the target data set name field because when this tool expands the data set name it only pick up the data to the left of the '/' code as the data name match search pattern.

6. If you need to copy the PDS member to a new or an existing sequential file, then you may use a 'CS' command to perform such function. Similarly, the 'MS' command can be used to move a PDS member to a new or an existing sequential file.

3.11 How to use the 'RING' command on the PANEL4 panel

1. You may enter a 'RING' command on the PANEL4 panel command line and type few 'S' codes to the selected members to build an Edit Ring.

2. On the same PANEL4 panel screen you may type a pair of 'SS' code instead of typing few 'S' codes to select several members to build the Edit Ring.

3. If the 'S' or 'SS' code is not used, then a PDS Member Selection panel will be displayed. You may either use the 'S' code to select few members or enter an 'ALL' command on that panel to select all PDS member to build the Edit Ring.

4. In the Edit Ring, press the PF3 key can save currently edited file and move the ring to the next file. Press the PF4 key can save the currently edited file and move the ring to the previous file. You may also use the 'N' (Next) or 'BA' (BAck) command to move the ring in forward or backward direction without saving the updated data in the edited file.

5. New PDS members can be added to the Ring by using the 'EDX' or 'RC' command. In the Edit Ring, you may use the 'G' (Get) and 'P' (Put) edit commands to cut and paste the data among several edited ring files very easily. Please see the Reference Guide for more details.

6. Enter a 'QQ' or 'OUT' command can get out of ring.

7. Enter a 'BRRING' command can build the Browse Ring.

8. Enter a 'VWRING' command can build the View Ring.

9. The RING command can also be used on the RECALL command panel which will be discussed next.

3.12 How to use the 'TRAP' command

1. The 'TRAP' command can be used to trap the REXX program trace output and display it on a VIEW command listing for you to review.

2. Enter a 'TRAP' command next to a file on the PANEL3 panel, next to a PDS member file on the PANEL4 panel, or next to a file on the ISPF option 3.4 data set list panel can display a window panel for you to enter a command name and optionally enter the parameter data to invoke the REXX program. If the select file is a REXX program, then you may enter a 'TRAP EXEC' command or type an 'EXEC' command name on the 'TRAPCMD' window panel. If the parameter data is 'TEST', then on the window panel you may enter an 'EXEC / TEST' to invoke the REXX program, where the '/' code stands for the selected data set name.

3. You may enter a 'FFF' command on any process panel to display a 'DSLISTF' panel and then select a code entry to display the ISPF option 3.4 data set list panel for the 'TRAP' command function. There are totally three code lines area on the 'TRAPCMD' panel available for you to enter data. Please press the PF1 key on that panel for more detail information.

4. Whenever the Enter key is pressed on the 'TRAPCMD' panel, the entered command or parameter code lines will be saved in a shared pool buffer automatically. Next time when you invoke the 'TRAP' command again, you may press the PF7 or PF8 key on this panel to retrieve the previously saved command or parameter code lines. To clear up all previously saved command or parameter code lines, simply press the PF10 key. Maximum up to 7 different command and parameter code lines can be saved in the buffer.

5. Be sure to set a 'Trace R' code in the REXX program before you start to trap the trace output.

6. The 'TRAP' command can be used to trap the trace output of the CLIST program also. On the trace output VIEW listing of the REXX program, you may position the cursor on a code line number and press the PF4 key, i.e. the 'ED' command key, to display the REXX program source in edit. The selected code line will be displayed on the top of the edited REXX program listing so that you may modify it very easily.

3.13 How to use the 'RECALL' command

1. Enter a 'RECALL' or 'RC' command on any process panel can display a RECALL command panel. Note that the 'RC' command works in edit file also.

2. The RECALL command panel contains a list of files in the sequential file or PDS member file level, which are the files that you have previously edited on either the PANEL3 or PANEL4 panel before.

3. On the RECALL command panel you may type a selection code, position the cursor at any column on a data entry, or type an 'E' code to select a file for edit. You may type a 'B' or 'V' code next to a file to browse or view the file also.

4. If you type an 'E' command next to a file which you are not authorized to update, then this tool will automatically convert the edit command function to the view command function unless an 'EDX' command is used. However, to use an 'EDX' command to edit an unauthorized file is not recommended.

5. On the RECALL command panel enter a 'RING' command can edit the files in an Edit Ring. You may use few 'S' codes or a pair of 'SS' code to select several files to build the Edit Ring on the RECALL command panel.

6. Enter a 'SW' (Swap) command can display the timestamps of the files on the RECALL command panel. From this timestamp information you may find out when was the last time you have previously edited a file.

7. The 'EDL' or 'EDLAST' command can be used on any panel, such as PANEL2 panel, to edit the last used file, which is usually displayed on the top of the RECALL command panel.

8. If you need to edit the second last or the third last edited file, then you may enter a 'EDL 2' or 'EDL 3' command instead of the 'EDL' command. However, in this case it is recommended to use the 'RC' command because the RECALL command panel is easier to use.

9. Enter a 'XX' command on the RECALL command panel can switch back to PANEL2 panel. Note that the 'XX' command can be used on any process panel to branch back to the PANEL2 panel.

10. Enter a selection code on the RECALL command panel can switch to another PANEL3 or PANEL4 panel. You may place the cursor on the Selection Code field or Project Name field on the RECALL command panel and press the Enter key to branch to the selected PANLE3 or PANEL4 panel also.

11. Instead of using cursor, you may use the '/a.b', 'a.b', or '/a' type of selection code, such as '/3.2', '5.4', '/3', etc. on the RECALL command panel to branch to the selected PANLE3 or PANEL4 panel from the RECALL command panel.

12. If no selection code is chosen when the Enter key is pressed, then the file on the top of the screen or the previously edited file on the RECALL command panel will be edited.

13. Use the 'D' command or a pair of the 'DD' command code can delete the file names from the data set name list on the RECALL command panel.

Note: The ISPF Workplace REFLIST panel function is very similar to the RECALL command function. On the REFLIST panel there are only few data set names you may select. Some of the data sets are displayed in the PDS file level. Thus, when you select the PDS on this panel you still have no idea which member in this file was most recently edited. The RECALL command function, on the other hand, not only has no panel size limitation but also has the timestamp associated to each recorded file that is in the sequential file and the PDS member file level. Therefore the RECALL command function is a lot better than the REFLIST function.

3.14 How to swap between two projects using '//' and 'BK' commands

1. On the PANEL3 or PANEL4 panel of any project you may enter a '//' command to display a Project Selection panel.

2. Use PF7 or PF8 key on that panel you may select a project. After pressing the Enter key it will branch to a new project.

3. On the new project enter a '//' command again. You will see that the original project to be shown on the displayed Project Selection panel.

4. Press Enter key can branch right back to the original project.

   Thus, the '//' command allows you to switch back and forth between two project or work items.

5. Use '//2', '//3', etc. can be used to switch to any other designated projects.

6. On the PANEL3 or PANEL4 panel, enter a '/a.b', 'a.b', or '/a' type of selection code, such as '/3.2', '5.4', '/3', etc. can branch to a new project. On the new project panel you may also enter a '//' command to display a Project Selection panel to branch back.

The '/a.b' command can always be simplified as the 'a.b' command. It can branch to the target 'a.b' project panel from the original panel on which you entered the 'a.b' command.

Other than using the '//' command to branch back, you may use a '??' command to display the selection code of the original panel. You may also use a 'BACK' or 'BK' command to branch back to the original panel immediately without displaying the selection code of the original panel. Thus, the 'a.b' and 'BK' command pair allows you to switch between two projects back and forth very easily.

For example, if the selection code '1.1' in your PROJWRK tool session is the 'PANEL(ISR@PRIM)' ISPF command of the sample '@MISC' project, then on any process panel you may enter a '/1.1' command to branch out the PROJWRK tool session and enter into the ISPF primary menu. On that menu you may select any function, such as DB2 SPUFI function, etc. that are not included in the PROJWRK tool functions. After that, you may enter a '=X' command on any ISPF panel to branch back to the PANEL3 panel of the '@MISC' project, and then enter a 'BK' command to branch back to the original process panel.

7. If you are on the '4.4' PANEL4 panel and you want to switch to '4.3' PANEL4 panel, instead of using a '4.3' or a '3' command, you may enter an 'A' (Above) command. If you want to switch to '4.2' PANEL4 panel, then you may enter an 'AA' command, etc. You may use a 'BK' command to switch back.

   Similarly, if you are on the '4.1' PANEL4 panel and you want to switch to '4.2' PANEL4 panel, instead of using a '4.2' or a '2' command, you may enter an 'N' (Next) command. If you want to switch to '4.3' PANEL4 panel, then you may enter an 'NN' command, etc. You may use a 'BK' command to switch back in this case also.

   Thus, the 'A..' and 'BK' command pairs and the 'N..' and 'BK' command pairs allow you to work on two PDS files back and forth very easily.

3.15 How to switch to another project or switch to another file

1. The 'XX' command can be used to branch back to PANEL2 panel from PANEL3, PANEL4, or RECALL command panel, or in edit file. On the PANEL2 panel you may select a new project.

2. The 'XRF' command can be used to display the PANEL5 panel from PANEL2, PANEL3, or PANEL4 panel. On the PANEL5 panel you may select a new data set.

3. The '/a' type of selection code, such as the '/1', '/2', etc. selection code on the PANEL3 or PANEL4 panel can be used to branch to a new project.

4. You may use the '/a.b' or 'a.b' type of selection code, such as the '/3.2', '5.4' etc. selection code, on the PANEL3 or PANEL4 panel to branch to a new PANEL4 panel.

5. The '/a.b', 'a.b', and '/a' type of selection code also applies to the RECALL command panel.

6. The 'b' type of selection code, such as the '4', '8', etc. selection code, can be used on the PANEL4 panel to branch to a new PANEL4 panel of the same project.

7. The '//', '//2', etc. command method can be used on the PANEL3 or PANEL4 panel to switch to a new project.

8. On the PANEL4 panel you may press PF3 to return back to the PANEL3 panel and then enter a selection code or use the cursor and press the Enter key to display another PANEL4 panel of the same

project.

3.16 How to use the 'FFF' command to replace the ISPF option 3.4

1. On any process panel enter a 'FFF' command can display the 'DSLISTF' panel, which is the front−end process panel of the ISPF option 3.4 data set list panel.

2. Originally this panel contains only two entries. One is your TSO Userid and the other is the Dsname Level code of 'SYS1'. On this panel, you may use the 'E' command to edit the 'DSLISTF' database and add more new Dsname Level code lines.

3. Select a Dsname Level code on the 'DSLISTF' panel and press Enter key, you can display the ISPF option 3.4 data set list panel of the selected Dsname Level code.

4. The 'FFF' command can be entered on the command line of the edited or viewed file also.

3.17 How to use the 'SRCH' command on the PANEL4 panel

1. Enter a 'SRCH' command next to a PDS on the PANEL3 panel or enter a 'SRCH' command on the PANEL4 panel command line can display a 'String Search−For Utility' panel to perform the String Search function on a single file.

2. By pressing the PF11 key on the 'String Search−For Utility' panel you can display a PDS Member Selection panel which allows you to select several members in PDS to be searched.

3. After the searched data string is found, the result will be displayed in a viewed listing.

4. On the viewed result listing you may use PF4 key to edit the member files that contain the searched data strings and perform the "Massive Change" function.

5. On the String Search function generated viewed result listing you may also enter a 'CP' command to copy the PDS member files found on the listing to a new created PDS. This new PDS will be created on your TSO account. On any process panel you may enter a 'FFF' command to display a 'DSLISTF' panel and then display the ISPF option 3.4 data set list panel of your TSO account. Then you may type a 'PP' command next to the new created PDS to print the entire PDS at a printer for your reference.

6. Enter a 'SRCH' edit command on the edit command line of an edited PDS member file and position the cursor in the file area to fetch the searched data string from the screen can also perform the String Search function.

7. The 'SRCH /' command can be used to re−display the latest 'String Search−For Utility' result listing. This command can be entered either on any process panel command line or on the edit command line.

8. On the 'String Search−For Utility' panel you may enter a 'B' command to submit a batch job to search for the string data from the selected data set. After the batch job is completed, you may use the 'SRCH /' command to display the result in a viewed listing format.

3.18 How to use the 'XSRCH' command on the PANEL3 or PANEL4 panel

1. Enter a 'XS' or 'XSRCH' command on the PANEL4 panel command line can display an 'Extended String Search−For Utility' panel to perform the Extended String Search function on a single file.

2. Enter a 'XS' or 'XSRCH' command on the PANEL3 panel command line and type few 'S' codes to select the data sets can display an 'Extended String Search−For Utility' panel to perform the extended string search function on multiple files. If the 'S' code is not specified, then all the files defined on the PANEL3 panel will be selected.

3. On the PANEL3 panel you may also type a 'XS' command next to a PDS to perform the extended string search function on a single file.

4. By pressing the PF11 key on the 'Extended String Search−For Utility' panel you can display a PDS Member Selection panel which allows you to select several members in PDS to be searched for the data string.

5. After the searched data string is found, the result will be displayed in a viewed listing.

6. On the result listing you may use PF4 key to edit the member files and perform the "Massive Change" function.

7. Enter a 'XS' or 'XSRCH' command on the RECALL command panel and use the 'S' code to select several files can perform the extended string search function.

8. Enter a 'XS' or 'XSRCH' edit command on the edit command line of an edited PDS member file and position the cursor in the file area to fetch the searched data string can also perform the extended string search function.

9. The 'XS /' or 'XSRCH /' command can be used to re−display the latest "Extended String Search−For Utility" result listing. This command can be entered either on any process panel command line or on the edit command line.

10. On the 'Extended String Search−For Utility' panel you may press the Enter key to proceed the search for data string function. You may also enter a 'B' command to submit a batch job to search for the string data from multiple data sets. After the batch search function is completed, you may use the 'XS /' command to display the result in a viewed listing format.

11. The PF4 key has been assigned as the 'ED' command function as the default. The 'ED' command is a very useful utility program which has been created to perform this very special function for editing the member file where the member file name can be fetched from the 'SRCH' or 'XSRCH' command generated result listings.

3.19 How to use the 'DIFF' command on the PANEL3 or PANEL4 panel

1. On the PDS Member List of the PANEL4 panel you may type a 'DIFF' command next to a member file. Then the ISPF option 3.13 SuperCE Utility panel will be displayed.

2. On the SuperCE Utility panel you may modify the target New DS name field manually and then press the Enter key.

3. On the PANEL4 panel if you want to compare two member files, then you may fill in the "User Data" field on the panel with the target member name before pressing Enter key. In this case, you don't need to modify the target New DS name on the SuperCE Utility panel.

4. The 'DIFF' command will display a 'DIFFPNL' Utility panel, i.e. the SuperCE Utility front−end panel on which may either modify the target New DS name field manually or press the PF11 key to display a 'SELPROJ' panel and select the target New DS name from that panel.

5. If the target is an existing file but it cannot be found on the 'SELPROJ' panel, instead of pressing the PF11 key to search for it, you may type the target file name partially on the panel ended with a '/' code and press the Enter key to expand the data set name and display the data set name list on a selection panel, which is named EXPDSNS panel. On that panel you may type one and only one 'S' code and press the PF3 key to select a target data set name.

6. Note that the '/' code can be inserted in the middle of the target data set name field because when this tool expands the data set name it only pick up the data to the left of the '/' code as the data name match search pattern.

7. On the SuperCE Utility panel you may press Enter key to proceed the two files comparison function. The result will be displayed in a viewed listing format.

8. The 'DIFF' command is applicable to the PANEL3 panel. You may use the 'S' code to select the target data set name on the PANEL3 panel if both of the source and target file names can be found on the same PANEL3 panel screen.

9. Similarly, if the source and target member files on the regular form PANEL4 panel that you want to compare are on the same PANEL4 panel screen, then you may type a 'DIFF' command next to the source member and type an 'S' (Select) command next to the target member on the panel to compare the two member files. In this case, you don't need to type the target member name in the "User Data" area on the source member name line.

10. For the short form PANEL4 panel, you may enter a 'DIFF' command on the panel command line and position the cursor on the source member name on the panel to display the SuperCE Utility panel and fill in the target new DS name on that panel.

11. If you need to compare a PDS member file listed on the RECALL command panel, then it is recommended to type an 'X' command next to that file to display a PANEL4 panel of the selected PDS and then use the 'DIFF' command on the PANEL4 panel to compare the file.

12. The 'DIFF' command is also applicable to the data set name displayed on the ISPF option 3.4 data set list panel, which can be displayed by using the 'FF' or 'FL' command on the PANEL3 panel.

13. The 'DIFF' command is applicable to compare two uncataloged files, or compare a cataloged file with an uncataloged file on the PANEL3 or PANEL4 panel.

14. The 'DIFF /' command can be used to re−display the latest "Extended SuperCE" result listing.

3.20 How to use 'XRF' command to display PANEL5 panel

1. On the PANEL2, PANEL3, or PANEL4 panel you may enter a 'XRF' command to display the PANEL5 panel.

2. On the PANEL5 panel you may select the previously edited, browsed, or viewed member and repeat

the same command function.

3. On the PANEL5 panel you may place the cursor on a code line of another project and press Enter key to switch to that project.

4. The PANEL5 panel is not a frequently used panel. However, if you cannot remember which project or work item that a data set name belongs to, then it is not a bad idea to use a 'F' (Find) command on the PANEL5 panel to search for the data set.

3.21 How to use 'XX' command to switch back to PANEL2 panel

1. On each process panel enter a 'XX' command can switch back to PANEL2 panel.

2. The 'XX' command works on the RECALL command panel and it also works in the edited file.

3.22 How to use the 'DEST' and 'SF' command

1. Enter a 'DEST' command on any process panel can edit the Target Destination code database file. This file contains the Target Destination code that are required by the 'SF' command to determine where the file should be sent.

2. Use 'SF' command next to a data set on the PANEL3 panel can display a Target Destination panel. After pressing the Enter key on this panel the selected file can be sent.

3. On the Target Destination panel, initially you need to fill in a target destination Node/Userid data. After the data is filled in, it will be saved in a target destination code database file.

4. Type a 'SF' command next to a file and use the '=' command code for the other files on the PANEL3 panel can send several files at the same time.

5. Use 'SF' command on the PANEL4 panel command line can send a PDS file also.

6. The 'SF' command and the '=' code can also be applied to the PDS member files on the PANEL4 panel.

7. On the PANEL3 panel you may also enter a 'SF' command and use the few 'S' codes next to several data sets or use a pair of 'SS' code to select several data sets to be sent.

8. Press the PF10 key on the Target Destination panel can display a Target Destination Selection panel, i.e. the GETDEST panel, to select a Node/Userid code. On that panel you may select a Node/Userid code or use an 'E' command to edit the target destination code database file and update the Node/Userid code lines.

9. You may also use PF11 key on the Target Destination panel to display a PDS Member List to select few members to be sent.

10. The 'SF' command can also be used to send an uncataloged file.

11. In the edit file you may enter a 'SF' command and use a pair of 'CC' line command to send a code segment of the edited file to the target destination.

12. If a source file is very large and you want to copy just part of the code lines from this file, then you may either type a 'V' command next to this file on PANEL3 panel or type a 'VW' command next to

this file on the ISPF option 3.4 data set list panel, which was displayed by using a 'FFF' command, to display a VIEW command listing and then use a 'SF' command on the VIEW listing panel with a pair of 'CC' line command to send a code segment to the target destination.

Similarly, in the VIEW command listing you may enter a 'CPY' command with a pair of 'CC' line command to copy the selected code segment of the large file to any new created sequential file.

13. You may press PF1 key on the Target Destination panel for more information about how to use the 'SF' command. You may also enter a 'HLP XMIT' command on any process panel to display a on−line tutorial panel of the 'XMIT' command. By comparing the two tutorials, you will find that how easy to use the 'SF' command than the 'XMIT' command.

3.23 How to use the 'SFX' command

1. On the PANEL3 panel or ISPF option 3.4 data set listing panel, next to a PDS or Sequential file you may type a 'SFX' command to duplicate the selected source library file at another MVS system. This command uses the JCL to XMIT and RECEIVE the source library file and duplicate at the target MVS system automatically.

   When the 'SFX' command is executed, a Target Node/Userid window panel will be displayed. On that panel you will need to specify a TSO account Node/Userid with the password. The password is a required field for you to duplicate the file at the target MVS system.

2. You may press the PF10 to display the Target Destination panel and select a TSO Node/Userid data entry from that panel.

3. Since you may duplicate the source file at only one target MVS system at a time, thus, the 'GETNICK' command function on the Target Destination panel has not been provided.

3.24 How to use the 'PP' and 'SP' commands

1. On the PANEL3 panel, next to a PDS you may type a 'PP' or 'SP' command to send the selected PDS to the target destination in a sequential file format. The 'PP' and 'SP' command method is very similar to the 'SF' and 'SFX' command functions. All of these four types of file transmit commands share with the same target destination code database file.

2. You may also use PF10 or PF11 key to display the Target Destination panel and PDS Member List panel.

3. The XRSTOR batch job file and the XRESTORE program code in the GUIDE library can be used to restore the sequential file back to PDS.

4. To restore the sequential file generated by the 'PP' command back to the PDS format, you may apply the 'GETPDS' command.

3.25 How to use the 'SFS' command

1. On the PANEL3 panel, next to a sequential file you may type a 'SFS' command to display the Target Destionation panel. This command will create a new sequential file with the 'Line Splitter' code added to each code line of the input file. This new file can be downloaded to the workstation and uploaded back to host.

2. The sequential file generated by the 'SFS' command can be restored to its original sequential file format by using a 'GETSEQ' command.

3.26 How to use the CMD command to display Command List panel

1. On any process panel except the regular form PANEL4 panel press the PF10 key can display a Command List panel.

2. On any process panel including the RECALL command panel enter a 'CMD' command can also display a Command List panel.

3. Use PF4 key on the Command List panel can display Command List database in a viewed listing format.

4. The source code of the Command List file can be found in the TABLE library. The User−Defined Command List file must reside in the XREF library.

3.27 How to use the LISTA and LISTC command

1. On any process panel you may enter a LISTA or LISTC command to display a LISTA command panel or a LISTC command panel.

2. On the LISTA or LISTC command panel, next to any file you may type an 'ED' command to edit the file, type a 'B' command to browse the file, or type a 'V' command to view the file.

3. On the LISTA or LISTC command panel, it is recommended to use the 'ED' command rather than the 'E' command to edit the file. Similarly, next to a data set on the ISPF option 3.4 data set list panel it is recommended to use the 'ED' command to edit a file also. The 'ED' command can display the PANEL4 panel on which it allows you to apply various kinds of utility commands provided by this tool package.

4. On the LISTA or LISTC command panel, you may enter a 'XSRCH' command and type few 'S' command to perform the Extended String Search function.

5. You may use the 'FM' (Find Member) command and type several 'S' code on the LISTA or LISTC command panel to find a member from a group of selected PDS files. The 'FM' command can be applied to the PANEL3 panel also.

6. The 'LINKLIST' and 'LPALIST' command can be used to display the Link List panel and the LPA List panel.

7. On each of these LISTA and LISTC command panels, you may use a 'DSN' command to generate a DSNLIST file.

3.28 How to use 'FM' command to search PDS member

1. On the PANEL3 panel you may enter a 'FM' (Find Member) command and type several 'S' code to find a member from a group of selected PDS files. If no 'S' code is specified, then all the PDS on the PANEL3 panel will be searched.

2. Similarly on the LISTA, LISTC, LINKLIST, or LPALIST command panel you may enter a 'FM' command and type several 'S' code to find a member from a group of selected PDS files.

3. On the ISPF option 3.4 data set list panel displayed from 'DSLISTF' panel you may use a 'FINDMEM' command without specifying the 'S' code to find the PDS members.

4. When the 'FM' command is executed, a 'FINDMEM' utility panel will be displayed. On that panel you may enter a 'B' command to submit a batch job to search for the PDS member. It is recommended to search for the PDS member in batch jobs because the foreground process is very slow.

5. After the 'FINDMEM' process is completed, you may use the 'FM /' command to display the "Find PDS Member utility" result listing in a viewed listing format.

6. On the result listing you may use the PF4 key to edited the member file that is found in the selected PDS.

7. If the 'FM' is to be used as an Edit Macro in the VIEW listing of the SRCHFOR output listing, which is generated by the 'SRCH' or 'XS' command, then all the members listed in the listing will be displayed in a PDS Member List panel. The option codes of the 'FM' edit command is 'IN', 'OUT', "INLIST', and 'OUTLIST'. The option code 'IN' is the default, which can be omitted.

3.29 How to use the User−Exit commands

1. You may use an 'U' command command to edit the User−Exit command database file. In that file, you may either add your own User−Exit commands or delete some existing User−Exit commands.

2. Use 'J' command next to a JCL file on the PANEL3 or PANEL4 panel can submit a batch job.

3. Use 'J' command on any process panel command line or next to a non−JCL file on the PANEL3 or PANEL4 panel can display the batch job output listing.

4. Use 'U J' command in edit can display the batch job output listing also.

5. Use 'REC' command on any process panel can receive the files.

6. Use 'U REC' command in edit can receive the files also.

7. Use 'HLP XMIT', 'HLP ALLOC', and 'HLP LISTA', etc. command can display the TSO command on−line tutorial guide in a panel format.

8. There are few other useful sample User−Exit commands defined in the User−Exit command database. For example, the '\11' command can be used for the ISPF Workplace function.

9. The source code of the '@UXSAMP' program and the 'REXCMP' program in the CEXEC library are good examples which can assist you to create your own User−Exit command routines in REXX. You may copy the '@UXSAMP2' program, which is a simplified version of the '@UXSAMP' program, from the CEXEC library to your program code to start to write your own User−Exit command routine.

3.30 How to use various Edit Macro commands in edit files

There are many Edit Macro commands have been provided by this tool package. In this section, several very useful Edit Macro commands have been described for your reference.

1. In edit you may use the 'T' or 'BOT' Edit Macro commands to branch to the top or bottom of the file.

2. In edit you may use an 'EDX' Edit Macro command to display the PANEL4 panel and use 'Z' command on the PANEL4 panel to compress the PDS without abort the modified edited file when the E37 problem occurs. To split the ISPF panel screen and compress the PDS is very tedious. This method is a lot superior.

3. The 'G' (i.e. Get or Cut) and 'P' (i.e. Put or Paste) command can be used to cut and paste the data among several edited files. Instead of entering a 'G' command on the edit panel command line area and a pair of 'CC' or 'MM' line command to copy or move the code lines and entering a 'P' command on the edit panel command line and and a 'A' or 'B' line command to paste the cut code lines. You may also type a 'G' or 'P' line command in the prefix line command area to cut or paste the code lines.

4. The 'SJ' Edit Macro command can be used as the extended split and join function, which can split or join the code line in edit file very easily. It is the short command form of the 'XSPLIT' Edit Macro. In edit the 'XSPLIT' function has been assigned to the PF2 key. When the cursor is put at the Edit command line area or prefix line command area when the PF2 key is pressed, then the original ISPF 'SPLIT' function to split the ISPF screen will be performed. Otherwise, the code line in the file area or comment box will be split or joined based on the cursor position.

   Similarly, the 'XSWAP' Edit Macro command can be used to swap the ISPF screen if it is in split mode. In edit the 'XSWAP' function has been assigned to the PF9 key. When the ISPF screen is not in the split mode, the 'XSWAP' Edit Macro will be automatically converted to the 'GETVAR' or 'PUTVAR' Edit Macro function.

5. The 'EXPAND' Edit Macro is a function that can be used to expand the partially specified data set name in edit into a full qualified data set name list and it allows you to choose the data set name from a selection panel so that you don't need to remember the data set names that you will define in the Project List member file. More detail information about the usages of this function can be found in the 'How to use EXPAND Edit Macro' section.

6. The 'CS' Edit Macro is the extended Column Shift function. It can be used to shift the column to the left or right just based on the cursor position in the file area.

7. There are several Edit Macro commands, such as 'AUTOINS', 'CMTBOX', 'CMTLINE', 'COLUTIL', 'SORTUTL', and 'NODUP', etc. can be used in edit files. Please see the Appendix B section for more information about the usages of these Edit Macro commands.

8. If you prefer to reset the PF2 key to the original ISPF 'SPLIT' function, simply enter a 'KEYS' command in edit to display a Keylist Utility panel and change the 'XSPLIT' command in PF2 key to the 'SPLIT' command. In this case, you may still keep the PF14 key as the 'XSPLIT' command function. To use the PF14 key, you may place the cursor on the code line and press the Shift key and PF2 key at the same time. Similarly, if you prefer to reset the PF9 key to the original ISPF 'SWAP' function, simply enter a 'KEYS' command in edit to display a Keylist Utility panel and change the 'XSWAP' command in PF9 key to the 'SWAP' command.

9. The 'CA' Edit Macro command can be used to branch to a Subroutine Label from a Call Statement (or a Signal Statement in REXX routine) and the 'RET' Edit Macro command can be used to return from the Subroutine Label back to the Call Statement.

10. You may use the 'FF', 'ALL', or 'FSTR' Edit Macro command to search for the data string. The 'FF' Edit Macro command requires to use the PF5 key to repeatedly search for the data string in the edit file. For all of these three commands, you may either specify the search data string as the command parameter or position the cursor on a non−blank code line in the file area to pick up the searched data text.

11. The 'BM' and 'GM' Edit Macro commands can be used to set and get the bookmarks in edit, and the 'BMQ' Edit Macro command can be used to display all bookmarks on a panel for easy selection.

12. The 'LABELS' Edit Macro command can be used to put a bookmark on each subroutine label in the edited file. After using this command, you may use the 'GM' or 'command to locate each subroutine one by one sequentially. Enter a 'GMQ' command can display a bookmark panel on which it allows you to branch to any subroutine very quickly.

13. The 'BM' and 'GM' are two different Edit Macro commands. However, the functions of the 'BMQ' and 'GMQ' Edit Macro commands are identical because both commands can be used to query the bookmarks and display them on a panel.

14. The 'RC' command in edit file can display a RECALL command panel which has been described previously before.

15. This tool has provided an Auto Bookmark function, which allows any edited file to be re−displayed just like the last edited session so that the users may very easily to continue the same file editing work even after many days or many months. If you don't like this feature, then you may use a 'TSO TOP OFF' command to drop this function.

16. The 'TAG' Edit Macro command can be used to add a program tag on all changed code lines in an edited program file. You may use a 'TAGS' edit command to display a table panel to define up to 12 frequently used program tags. If the edied file is a script file, then you may use a 'TAG' line command to insert a GML tag at column 1 of a script code line.

17. The 'PWOPT' command can be used to display an "Options Selection" panel on which the 'TSO TOP' command can be set to 'ON' or 'OFF'. To drop the Auto Bookmark function is not recommended.

In next few sections, several frequently used Edit Macro commands will be described. More information about the examples of the usages of these Edit Macro commands will be included in the section of 'The Edit Macro command functions' of this document.

3.31 How to use the XSPLIT Edit Macro

1. The 'XSPLIT' command is the extended Split and Join Edit Macro command. The short form of the 'XSPLIT' command is the 'SJ' command.

2. In edit you may enter a 'SJ' command and position the cursor on the code text and press the Enter key to split or join the code line.

3. The original ISPF 'TS' line command can perform only one part of the 'XSPLIT' command function. It cannot be used to join the code lines.

4. The 'XSPLIT' command has already been pre−assigned to the PF2 key command function in edit. To split or join the code lines, all you need to do is to position the cursor on the code text in the file area and press the PF2 key.

5. If the cursor is at the Edit command line area or at the prefix line command area when the PF2 key is pressed, then the original ISPF screen split function will be performed.

6. This Split/Join function also applies to the comment line in a comment box. (The CMTBOX Edit Macro command, which can be used for inserting the comment boxes in edit, will be presented in this section).

7. You may enter the 'XSPLIT ?' or 'SJ ?' command in edit for more detail information about the XSPLIT Edit Macro.

3.32 How to use the XSWAP Edit Macro

1. The 'XSWAP' command is the extended Swap Edit Macro command. It has already been pre−assigned to the PF9 key command function in edit.

2. If the ISPF screen is in the split mode, then in edit you may position the cursor at any location and press the PF9 key to swap the screen.

3. If the ISPF screen is not in the split mode, then the 'XSWAP' command will be automatically converted to the 'GETVAR' or 'PUTVAR' Edit Command function depending upon where the cursor is placed. If the cursor is placed at a non−blank area in the edit file area, then it will perform the 'GETVAR' Edit Command function. Otherwise, it will perform the 'PUTVAR' Edit Command function.

4. You may enter the 'XSWAP ?' command in edit for more detail information about the XSWAP Edit Macro.

3.33 How to use the COLSHFT Edit Macro

1. The 'COLSHFT' command is the extended Column Shift Edit Macro command. The short form of the 'COLSHFT' command is the 'CS' command.

2. In edit you may enter a 'CS' command and position the cursor on a code text and press the Enter key to shift a single code line to the left or right based on the cursor position. If the cursor is placed to the left of the leftmost non−blank character of the code text, then the columns will be shifted to the left. Otherwise, the columns will be shifted to the right.

3. You need to enter a pair of 'CC' line command to select a block of code lines to the left or right by using the 'CS' command. In this case, the cursor column position will be compared with the leftmost non−blank character of all code lines in the block to determine if the columns will be shifted to the left or right.

4. Instead of entering a 'CS' command on the edit panel command line area, you may type a 'CS' line command in the prefix line command area and use the cursor to control how many columns to shift the code left or right. You may also use a pair of 'CSS' or '//' line command to shift a block of code lines.

3.34 How to use the AUTOINS Edit Macro

1. In edit you may enter an 'I' command and position the cursor in the file to enter into the Automatic Insertion Mode. The PF4 key in edit has been assigned to this command function.

2. Use PF4 key you can enter into Automatic Insertion Mode very easily. The AUTOINS mode contains the Balance Code Insertion function and the Cursor Intelligent Movement function.

3. If the edited file is a JCL file, when you place the cursor on a code line and press PF4 key, if the code line contains a 'DSN=' keyword, then this tool will automatically edit the file specified in the 'DSN=' code. Otherwise it will insert a blank JCL code line with '//' code at the leftmost two columns.

4. In edit enter an 'IO' command can display the AUTOINS Option Selection and Balance Code Profile Group panels for you to make a choice on how to control the automatic code insertion function in edit.

5. You may enter the 'AUTOINS ?' or 'I ?' command in edit for more detail information about the AUTOINS Edit Macro.

6. Instead of entering an 'I' command on the edit panel command line area, you may type a 'AI' line command in the prefix line command area to enter the 'Automatic Insertion' mode in edit.

   The 'AI' line command is applicable to the SCRIPT file also. In an edited SCRIPT file, if a '

   ```
   ' tag is typed in, then 'AUTOINS' program
   will automatically insert a '
   ```

' tag at next script code line.

3.35 How to use the CMTBOX Edit Macro

1. In edit you may enter a 'BX' command and position the cursor in the file to insert an empty comment box.

2. The 'BXF' command can be used to format the context within a comment box.

3. In edit enter a 'BXO' command can display the CMTBOX Option Selection panel for you to make a choice on how to control the insertion of the comment box in edit file.

4. You may split and join the context in a comment box by using the 'BX' command.

5. You may enter the 'CMTBOX ?' or 'BX ?' command in edit for more detail information about the CMTBOX Edit Macro.

6. Instead of entering an 'BX' command on the edit panel command line area, you may type a 'BX' line command in the prefix line command area to insert a comment box.

3.36 How to use the CMTLINE Edit Macro

1. In edit you may enter a 'CL' command and position the cursor in the file to insert a remark comment line.

2. In edit enter a 'CLO' command can display the CMTLINE Option Selection panel for you to make a choice on how to control the insertion of the remark comment line in edit file.

3. You may enter the 'CMTLINE ?' or 'CL ?' command in edit for more detail information about the CMTLINE Edit Macro.

4. Instead of entering an 'CL' command on the edit panel command line area, you may type a 'CL' line command in the prefix line command area to insert a comment box.

3.37 How to use the COLUTIL Edit Macro

1. In edit you may enter a 'COL' command and use a pair of 'CC' prefix line command to define a range of code lines.

2. You may use the 'I', 'D', 'R', 'B', etc. command code on the horizontal command bar to manipulate the data in the selected column zones.

3. You may enter the 'COLUTIL ?' or 'COL ?' command in edit for more detail information about the COLUTIL Edit Macro.

4. Instead of entering an 'COL' command on the edit panel command line area, you may type a pair of 'CXX' line command in the prefix line command area to insert a comment box.

3.38 How to use the SORTUTL Edit Macro

1. In edit you may enter a 'SRT' command and use a pair of 'CC' prefix line command to define a range of code lines.

2. You may use the 'A' or 'D' command code on the horizontal command bar to sort the data in the ascending or descending order in the selected column zones.

3. You may enter the 'SORTUTL ?' or 'SRT ?' command in edit for more detail information about the SORTUTL Edit Macro.

3.39 How to use the NODUP Edit Macro

1. In edit you may enter a 'NODUP' command and use a pair of 'CC' prefix line command in a sorted file to define a range of code lines.

2. You may use the 'N' command code on the horizontal command bar to define the no duplication column zones.

3. You may enter the 'NODUP ?' command in edit for more detail information about the NODUP Edit Macro.

3.40 How to use the DIR command

1. On any process panels enter a 'DIR' command can display the Project Work Directory panel, i.e. the PROJDIR panel.

2. On the PROJDIR panel enter a 'NEW' command can create a Project List pair of the secondary Project Work Manager tool session.

3. On the PROJDIR panel you may enter a team member's Userid to access other person's Project Work Manager tool session.

3.41 How to use the short form PANEL4 panel

1. On the regular form PANEL4 panel enter a 'SW' (SWap) command can swap the displays between the regular form PANEL4 panel and the short form PANEL4 panel.

2. The 'SW' command can also be entered on the PANEL2 or PANEL3 panel to set the PANEL4 panel in the short form mode.

3. On the short form PANEL4 panel you may position the cursor on a member name and press the Enter key to select a PDS member.

4. After few PDS member files are edited, you may use '\' command to backtrace each member file in a certain sequence.

5. Use the '\\' command can display a PDS members backtrace panel. On that panel you may select any member without needing to worry about the backtrace sequence.

6. On the short form PANEL4 panel you may press the PF10 key or enter a 'CMD' command to display the Command List panel. On the regular form PANEL4 panel you may only enter a 'CMD' command to display the Command List panel because the PF10 is the 'Action' function PF key.

7. Press the PF1 or the PF11 key can display the tutorial of the short form PANEL4 panel.

8. The short form PANEL4 is not a frequently used panel because it is not as easy as the regular form PANEL4 panel to use. However, when you deal with a PDS with a lot of members, then it is not a bad idea to use this panel to save your time when searching for the member names on the panel.

3.42 How to view the LOG.MISC file from a project and switch back

1. On any process panels, say PANEL2 panel, you may enter a 'REC' User−Exit command to receive the files. The received file names will be automatically recorded in the LOG.MISC file.

2. On the PANEL3 or PANEL4 panel you may enter a '/1.2' or simply '1.2' command to display the LOG.MISC file in viewed listing format, assuming that the selection code 1.2 contains a 'VIEW LOG.MISC' command in the sample '@MISC' project, which is assumed to be the first project defined on the PANEL2 panel.

3. On the viewed LOG.MISC file listing, you may position the cursor on a received file name and press the PF4 key to display that file on the ISPF option 3.4 data set list panel.

4. Press PF3 on the viewed LOG.MISC file listing can close the file and return to the PANEL3 panel.

5. On the PANEL3 panel of the '@MISC' project enter a '//' or 'BK' command can branch back to original project.

6. Both of the '//' and 'BK' command methods are applicable to any of the '/a.b' type of commands on the PANEL3 and PANEL4 panels.

3.43 How to use the CPY command

1. On the ISPF option 3.4 Data Set List panel, which is displayed by using the 'FFF' command, you may type a 'CPY' command to display a window panel on which you may specify a target file name so that the selected file can be copied to the target file.

2. Instead of press the Enter key on the window panel, you may press the PF11 key to display a Data Set List panel to select the target file name.

3. The target file can be either a new file or an existing file. If it is a new file, then another window panel will be displayed to allow you to enter the attribute data of the new target file.

4. On the new target file attribute data panel, you may press the PF11 key instead of the Enter key to create an empty new target file without copying the source file, which is very much like the ISPF option 3.2 function.

5. Instead of using the PF11 key on the new target file attribute data panel to create a new empty target file, you may use a 'CRE' command to create a new empty target file based on the attribute of the source file just like the ISPF option 3.2 function.

6. If the 'CPY' is to be used as an Edit Macro in the VIEW listing of the SRCHFOR output listing, which is generated by the 'SRCH' or 'XS' command, then all the members listed in the listing will be copied to a newly created PDS.

3.44 How to use the HEXT command

1. On the PANEL3 panel or next to a PDS member file on the PANEL4 panel enter a 'HEXT' command can display the hexadecimal type of code listing of the selected input file. The name of the HEXT command generated output file is 'USERID.@HEXTYPE.DATA'.

2. On any process panel you may enter a 'HEXT /' command to display the previously HEXT command generated file on the ISPF option 3.4 data set listing panel. On that panel, you may type a 'V' command to view the file, or type a 'R' command to rename it to another file name, such as 'USERID.@HEXTYPE.DATA2' or 'USERID.@HEXTYPE.DATA3', etc.

3.45 How to use 'CPYFROM' command to copy a large file partially

1. The 'CPYFROM' command can be used to copy a part of the very large source file to a new created sequential file. The short form of the 'CPYFROM' command is the 'CPYF' command.

2. You may type a 'CPYF' command next to a very large sequential file on the PANEL3 panel, or next to a very large PDS member file on the PANEL4 panel, or next to a very large sequential file on the ISPF option 3.4 Data Set List panel, which is displayed by using a 'FFF' command, to display a window panel on which the start line number and the total lines for the copying function will be shown. You may specify the data at these two input fields on the window panel to determine where to copy the file and how many lines will be copied to the target file.

3. If these two fields are not filled in with data, then the default is to copy the entire source file to the target file.

4. On the window panel if the Enter key is pressed, then a batch job will be submitted to perform the file copying function. If the PF4 key is pressed, then the file copying function will be performed in foreground.

3.46 How to use 'HEXTFROM' command to generate the Hex code file

1. Similar to the 'CPYFROM' command, the 'HEXTFROM' command can be used to copy a part of the very large source file to a new created sequential file in the hexadecimal code format. The short form of the 'HEXTFROM' command is the 'HEXTF' command.

2. You may type a 'HEXTF' command next to a very large sequential file on the PANEL3 panel, or next to a very large PDS member file on the PANEL4 panel, or next to a very large sequential file on the ISPF option 3.4 Data Set List panel, which is displayed by using a 'FFF' command, to display a window panel on which the start line number and the total lines for the copying function will be shown. You may specify the data at these two input fields on the window panel to determine where to copy the file and how many lines will be copied to the target file.

3. If these two fields are not filled in with data, then the default is to copy the entire source file to the target file in hexadecimal code format.

4. On the window panel if the Enter key is pressed, then a batch job will be submitted to perform the file copying function. If the PF4 key is pressed, then the file copying function will be performed in foreground.

More information about the examples of the usages of the 'CPYF' and 'HEXTF' commands will be described in the details later in this document.

3.47 How to use the CAL command

1. On any process panel enter a 'CAL' command can display the current year calendar.

2. On the calendar you may press Enter key twice to display a Things−To−Do panel.

3. You may press the PF11 key to display a Notebook panel.

4. On the calendar you may use the PF5 or the PF6 key to display the Things−To−Do listing or the Notebook listing.

5. You may press the PF1 key and see the on−line tutorial for more information about the Notebook and Things−To−Do functions.

3.48 How to use the CALC command

1. On any process panel enter a 'CALC' command can display the Desktop Rolling Sheet Scientific Calculator.

2. On the calculator screen you may use the PF4 key to display different calculation mode.

3. You may press the PF1 key and see the on−line tutorial for the usages of the memory registers and various calculate functions.

3.49 How to use the FMT command to format the REXX source code

1. On the PANEL4 panel you may type a 'FMT' command next to a REXX program to format its source code.

2. On any process panel you may enter a 'FMT OPT' command to display the REXX Source Code Formatter options panel for you to make choice on how to control the REXX source code formatting.

3. The 'FMT' command can also be used in an edit file of a REXX program to format the source code.

4. In edit you may enter a 'FMT' command and a pair of 'CC' code to format a segment of REXX code.

5. Instead of using a 'FMT' command with a pair of 'CC' code, you may simply use a pair of 'FXX' line command to format a segment of REXX code.

3.50 How to use the REXXCHK command

1. On the PANEL4 panel you may type a 'REXXCHK' command next to a REXX program to check its syntax. After the 'REXXCHK' command is executed, the result listing will be displayed.

2. If any error is detected, then on the displayed result listing you may place the cursor on an error code line number on the listing and press the Enter key or PF4 key to edit the REXX program which contains the error code line on the top of screen.

3. Then you may fix the error in edit file and repeat the 'REXXCHK' command process.

3.51 How to use the REXXREF command

1. On the PANEL4 panel you may type a 'REXXREF' command next to a REXX program to display the REXX source reference listing.

2. On the displayed result listing you may place the cursor on an error code line number and press the Enter key or PF4 key to edit the REXX program which contains the error code line on the top of screen.

3. Then you may fix the error in edit file and repeat the 'REXXREF' command process.

4. The PF4 key has been assigned to the 'ED' command function. The 'ED' command is a very useful utility program which has been created to perform this very special function for editing the REXX source program with the error code line displayed on the top of the screen, where the error code line number is spotted from the 'REXXCHK' or 'REXXREF' command generated result listing.

Appendix A. The usages of the user−defined PROFILE file

Some information about the usages of the user−defined PROFILE file has already been described in the Appendix section in the Reference Guide. In this section, a sample user−defined PROFILE file will be presented again to describe the usages of this file in more detail.

A.1 The sample PROFILE member file

On any process panel of this tool package, such as the PANEL2 panel, if you enter a 'PROF' command then an empty PROFILE file will be created. The following is an example of such file, in which some library file names have already been defined for your reference. Please see the prologue section in this file for the information about the usages.

Figure 1. The sample user−defined PROFILE file

```
/*-----------------------------------------------------*/
/* THE USER PROFILE TO DEFINE LINKED FILES IN SETUP  */
/*-----------------------------------------------------*/
/* This PROFILE contains the control data to define  */
/* the linked files during the bring-up process. The */
/* valid control data are 'YES' and 'NO' in the      */
/* FILEAID and USEREXIT fields. In the other few     */
/* types of options, you may fill in the file names  */
/* of the EXEC, CLIST, SKELS, PANELS, LOAD, MSGS,    */
/* TABLE, HELP, etc. keyword entries that you wish   */
/* to link those library files to be interface with  */
/* this tool. If you wish those library files to be  */
/* accessible globally by all the users of this tool */
/* package, then you should check the file in the    */
/* 'TOOLKIT.@PROJWRK.CEXEC(@PROFILE)' for more      */
/* information. The names of the file must be fully  */
/* qualified without bounded single quotes, and they */
/* must be filled in after the equal sign of the     */
/* proper keyword code entry. The keyword code must  */
/* not be altered. Except the TABLE entry which can  */
/* have only one file, all other entries cannot have */
/* more than 15 files on each keyword entry.         */
/*-----------------------------------------------------*/
FILEAID  = NO   /* Default no File-AID interface
USEREXIT = NO   /* Default not to link User-Exit Lib
/* EXEC     = USERID.@USREXIT.EXEC
CLIST    = USERID.CODEPRTX.CEXEC
CLIST    = USERID.XEDIT.CLIST
SKELS    =
PANELS   = USERID.XEDIT.ISPPLIB
PANELS   = USERID.TASID.PANELS
LOAD     = USERID.TASID.LOAD
MSGS     =
TABLE    = USERID.XEDIT.ISPTLIB
```

Note:

1. The PROFILE file is a member in your 'USERID.@PROJWRK.XREF' library and it is displayed on

the screen in edit mode for you to update the control data, where the 'USERID' stands for your own TSO logon userid.

2. If you already have older version PROFILE member file in your XREF library, then please delete it first and then enter a 'PROF' command on the PANEL2 panel to create a new PROFILE file.

3. After you update the PROFILE member file, you will need to re−enter the PROJWRK tool session in order to make the changed options work. For example, if you change the USEREXIT option from 'NO' to 'YES', then you may enter a 'QQ' command at any process panel to exit the PROJWRK tool session first, then type a 'TSO PROJ' command to enter the PROJWRK tool session in order to access the command defined in your own User−Exit library, i.e. the 'USERID.@USREXIT.EXEC' file.

4. The 'FILEAID' option is for the control link to the File−Aid tool package library files. The File−Aid tool package is a Vendor Company developed software product. More detail information about this tool package can be found in the Reference Guide.

A.2 The descriptions of the sample PROFILE file

Suppose you have created a 'USERID.@USREXIT.EXEC' library file and would like to use the command defined in this file while you are in the PROJWRK tool session, then all you need to do is to change the control code of the 'USEREXIT' option in the PROFILE from 'NO' to 'YES'.

Other than using a control code 'YES' in the 'USEREXIT' option, you may explicitly specify that file name next to the 'EXEC' keyword entry, which is as shown in the above sample file.

In this example, the 'EXEC' keyword entry has been commented out with a '/*' code, which means that the 'USERID.@USREXIT.EXEC' file will not be concatenated to the other allocated 'SYSEXEC' libraries of this tool package during the initial bring−up. You may either delete the '/*' code in this file manually or position the cursor on that code line and press the PF4 key, i.e. the 'ED' command, to uncomment out the '/*' code. Note that the PF4 key can be used to toggle between the insertion and the removal of the '/*' code on a code line in edit file.

If you find any MVS tool package that are very useful and wish to use it, then you may very easily access it in the PROJWRK tool session simply by adding the tool library file names in your own user−defined PROFILE file, which is shown in the above example.

By using this 'PROF' command function, whenever you add a new library file name in the PROFILE file you can use it immediately after the PROFILE file is closed. Whenever you comment out a library file in the PROFILE file, however, it will not be removed from the system until you enter the PROJWRK tool session next time.

Appendix B. The Edit Macro command functions

In this section, the detail descriptions of the usages of several very useful Edit Macro commands, which have been provided by this tool package, will be presented in the diagram format.

B.1 The descriptions of the 'XCUT' and 'XPASTE' edit commands

The XCUT and XPASTE are the Edit Macro commands which can be used for the copy or move the code lines from one location to another in either the same or different edit files. These two edit commands are the command pair and they are the most useful Edit Macro commands provided by this tool package. The short forms of these two edit commands are the 'G' (Get) and 'P' (Put) commands, respectively.

B.2 The examples of the 'XCUT' and 'XPASTE' edit commands

The following few diagrams can describe how to use the "XCUT' and 'XPASTE' edit commands in the edit file. I personally prefer to use the 'G' and 'P' edit commands because they are a lot easier to use. Thus, in the sample diagrams, all the edit commands to be used for the Cut and the Paste functions are in the 'G' and 'P' command formats.

The following few diagrams can describe how the Cut and Paste functions work in edit file. Assume a 'G' edit command is entered at the command line and a pair of 'CC' line commands are typed in on line 3 and line 7, which is as shown below:

Figure 2. The method of entering a 'G' edit command in edit file

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(HELLO)                    Columns 00001 00072    |
|  Command ===> _g_____________________________________ Scroll ===> CSR        |
|  ****** ***************************** Top of Data ************************     |
|  000001  /* REXX */                                                           |
|  000002  Arg parm .                                                           |
|  0cc003  If parm = '?' then            <===                                   |
|  000004     Do                                                                |
|  000005        Call help                                                      |
|  000006        Exit                                                           |
|  000cc7     End                        <===                                   |
|  000008  If parm = 'SAY' then                                                 |
|  000009     Do                                                                |
|  000010        Say "Goodbye old world"                                        |
|  000011        Say "Hellow NEW world"                                         |
|  000012     End                                                               |
|  000013                                                                       |
+-------------------------------------------------------------------------------+
```

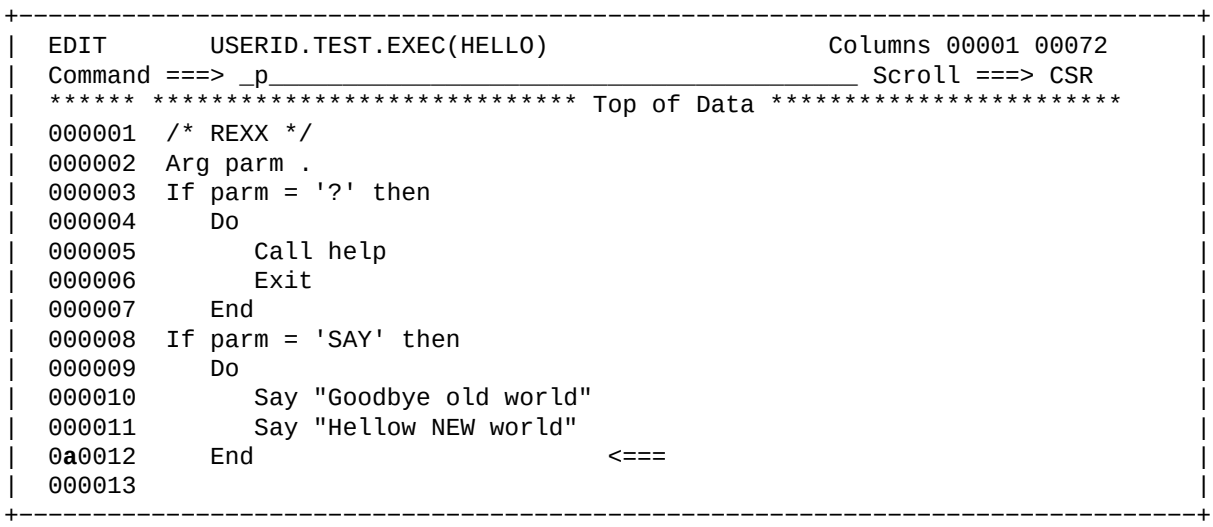
Note:

1. Instead of using a pair of 'CC' line command, you may use a pair of 'MM' line command to move the code lines from line 3 to line 7 to the target location in the same edit file or another file.

2. To copy or move a single code line, you just need to use a 'C' or 'M' line command instead of using a pair of 'CC' or 'MM' line command.

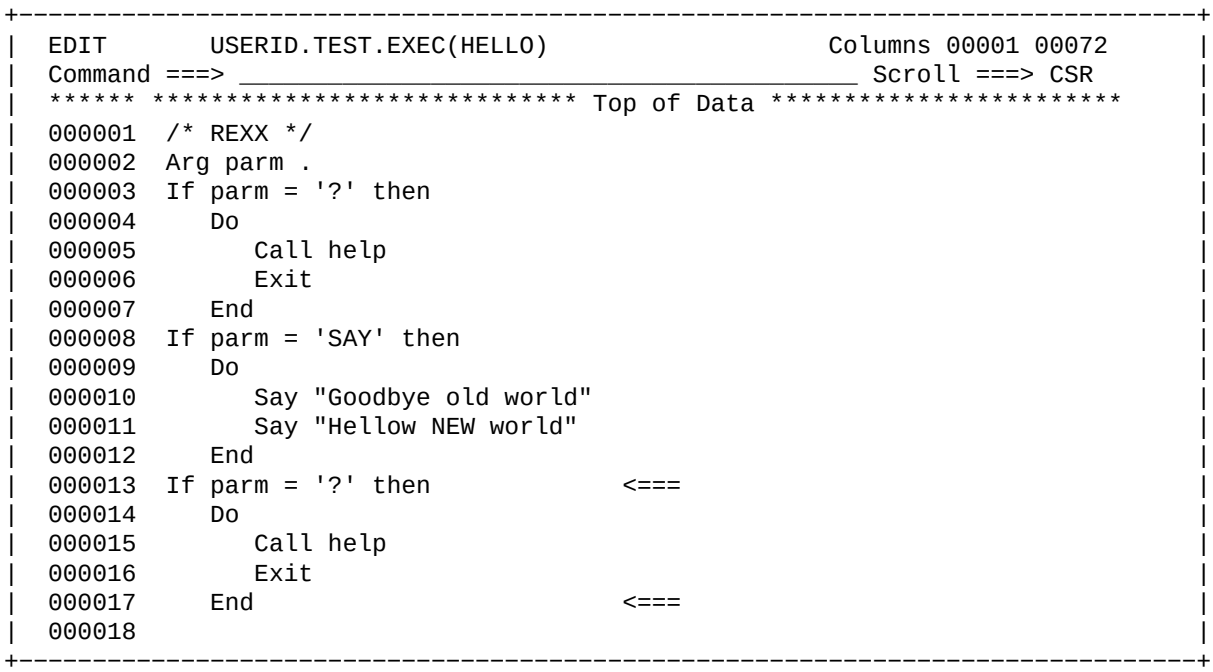
After pressing the Enter key, five code lines from line 3 to line 7 will be saved in a buffer. Now you may enter a 'P' edit command and then type an 'A' (After) line command at line 12, which is as shown below:

Figure 3. The method of entering a 'p' edit command in edit file



After pressing the Enter key, the five Cut lines will be pasted below the line 12, which is shown as follows:

Figure 4. The result of the 'p' edit command pasted code listing



Note:

1. After the 'P' edit command is executed, the buffer will be empty and you cannot perform the paste function of the same saved code again. Thus, instead of using the 'P' edit command you may use a 'K' option, where 'K' stands for 'KEEP', in the 'XPASTE' edit command to keep the saved code in the buffer for the multiple pastes. The short form of the 'XPASTE K' edit command is the 'P K' or 'PK' edit command.

2. If you continue to use 'G' edit command to cut the code lines before using the 'P' edit command, then the new cut lines will be appended to the end of buffer. In this case, you may copy or move the code

lines at several different locations to the target location.

3. Before using a 'P' edit command to paste the code lines, if you change mind and want to clean the buffer and restart the cut function, then you may use a 'C' option, where 'C' stands for 'CLEAR', in either the 'XCUT' or 'XPASTE' edit command. The short form of the 'XCUT C' and 'XPASTE C' edit commands is the 'G C' and 'P C' edit commands, respectively.

B.3 The descriptions of the 'XSPLIT' edit command

The XSPLIT is an Edit Macro command which can be used for the extended edit screen text data Split and Join functions. The short form of the 'XSPLIT' edit command is the 'SJ' edit command. The PF2 key in edit has already been assigned to this command function.

Instead of using the 'XSPLIT' or 'SJ' edit command to split or join the code lines in edit, it is highly recommended that you should use the PF2 key to split or join the data because the PF2 key is a lot easier to use.

If the cursor is at the ISPF edit command line area or the prefix line command area, i.e. the line number area, when the PF2 key is pressed, then the ISPF edit screen will be split. If the cursor is in the file area or in a comment box when the PF2 key is pressed, then the code line will either split into two lines or join with the next code line depending upon where the location of the cursor is. Please enter a 'SJ ?' command in edit for more detail information about this command.

The 'XSPLIT' (The extended text Split and Join) edit command function is a combination of the 'TS' (Text Split) line command and the 'C−O' (Copy override) line command functions. By using these ISPF line commands, you not only need to type in the commands at the prefix line command area, but also need to either shift the code lines to the right or manipulate the cursor before hitting the Enter key, which is very troublesome. By using this 'XSPLIT' command method, all you need to do is place the cursor at the proper position in the file area and press the PF2 key. This 'XSPLIT' command is smart enough to split and join the code lines in a comment box, which is the function that is not supported by the ISPF line commands.

B.4 The examples of the 'XSPLIT' edit command

The following few diagrams can describe how the "XSPLIT" function works in edit file. Assume a 'SJ' edit command is entered at the edit command line area and the cursor is placed under the first character '**t**' of the code '**then**' at line 7, which is as shown below:

Figure 5. The method of entering a 'sj' edit command in edit file

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(XSPLIT)                 Columns 00001 00072      |
|  Command ===> _sj_____________________________________ Scroll ===> CSR        |
|  ****** ***************************** Top of Data ***********************     |
|  000001  /*-------------------------------------------------------------*/     |
|  000002  /* XSPLIT - The extended Split/Join function for Edit Macro.   */     |
|  000003  /*-------------------------------------------------------------*/     |
|  000004  'ISREDIT MACRO (PARM)'                                               |
|  000005  If rc ^= 0 then                                                      |
|  000006     Exit                                                              |
|  000007  If parm = '?' length(parm) = 1 then       <===                  |
|  000008     Do                                                                |
|  000009        Call help                                                      |
|  000010        Exit                                                           |
|  000011     End                                                               |
|  000012  'ISREDIT (CURROW,CURCOL) = CURSOR'                                   |
```

```
|  000013                                                                    |
+----------------------------------------------------------------------------+
```

Note: In the above sample diagram, instead of using the 'SJ' edit command and manipulate the cursor, it is highly recommended that you may directly place the cursor at the same location on line 7 and press the PF2 key.

After pressing the Enter key, the code line 7 will be split into two lines and the word 'then' will be split into next code line. A new line will be inserted in between the two split lines for you to enter more input data and the cursor position will be remained unchanged, which is as shown below:

Figure 6. The selected cursor code line is split into two lines

```
+----------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(XSPLIT)                 Columns 00001 00072   |
|  Command ===> _____________________________________ Scroll ===> CSR        |
|  ****** **************************** Top of Data ***********************    |
|  000001  /*-------------------------------------------------------------*/   |
|  000002  /* XSPLIT - The extended Split/Join function for Edit Macro.   */   |
|  000003  /*-------------------------------------------------------------*/   |
|  000004  'ISREDIT MACRO (PARM)'                                            |
|  000005  If rc ^= 0 then                                                   |
|  000006     Exit                                                           |
|  000007  If parm = '?' length(parm) = 1 _            <===              |  |
|  ......                                                                    |
|  000008  then                                          <===                |
|  000009     Do                                                             |
|  000010        Call help                                                   |
|  000011        Exit                                                        |
|  000012     End                                                            |
|  000013  'ISREDIT (CURROW,CURCOL) = CURSOR'                                |
|  000014                                                                    |
+----------------------------------------------------------------------------+
```

If the PF2 key is pressed again when the cursor position is not moved, then the split word 'then' on line 8 will be joined back to the end of line 7 and the original code will be restored. If you delete the code **"length(parm) = 1"** on line 7 and then press the PF2 key, the new formatted code line 7 will look as the following:

Figure 7. The result of the code lines after the join process

```
+----------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(XSPLIT)                 Columns 00001 00072   |
|  Command ===> _____________________________________ Scroll ===> CSR        |
|  ****** **************************** Top of Data ***********************    |
|  000001  /*-------------------------------------------------------------*/   |
|  000002  /* XSPLIT - The extended Split/Join function for Edit Macro.   */   |
|  000003  /*-------------------------------------------------------------*/   |
|  000004  'ISREDIT MACRO (PARM)'                                            |
|  000005  If rc ^= 0 then                                                   |
|  000006     Exit                                                           |
|  000007  If parm = '?' then                                                |
|  000008     Do                                                             |
|  000009        Call help                                                   |
|  000010        Exit                                                        |
|  000011     End                                                            |
|  000012  'ISREDIT (CURROW,CURCOL) = CURSOR'                                |
|  000013                                                                    |
+----------------------------------------------------------------------------+
```

Assume that you have already typed in many words other than the word 'then' on line 8, then after the PF2 is pressed on line 7 some words on line 8 might be left behind and spill over into next line when the join data process is completed.

In the above sample diagram, if the cursor is placed in the middle of a comment box and it is under the first character '**e**' of the word '**extended**' in line 2 when the PF2 key is pressed, then that comment line will be split into two comment lines, which is as shown below:

Figure 8. The method of splitting the code in a comment box

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(XSPLIT)                 Columns 00001 00072      |
|  Command ===> _____________________________________ Scroll ===> CSR           |
|  ****** *************************** Top of Data ***********************      |
|  000001  /*-------------------------------------------------------------*/     |
|  000002  /* XSPLIT - The _                                              */     |
|  000003  /* extended Split/Join function for Edit Macro.               */     |
|  000004  /*-------------------------------------------------------------*/     |
|  000005  'ISREDIT MACRO (PARM)'                                               |
|  000006  If rc ^= 0 then                                                      |
|  000007     Exit                                                              |
|  000008  If parm = '?' then                                                   |
|  000009     Do                                                                |
|  000010        Call help                                                      |
|  000011        Exit                                                           |
|  000012     End                                                               |
|  000013  'ISREDIT (CURROW,CURCOL) = CURSOR'                                   |
|  000014                                                                       |
+-------------------------------------------------------------------------------+
```

Note:

1. If the cursor is placed on the edit command line area or the prefix line command area when the PF2 key, i.e. the 'XSPLIT' command key, is pressed, then the edit screen will be split just like the original regular 'SPLIT' command function.

2. This tool has setup the PF2 key as the 'XSPLIT' command as default. On the edit command line you may enter a 'KEYS' command to display a KEYLIST utility panel and change the PF2 key back to the 'SPLIT' command function. By doing so, you will not only eliminate the 'XSPLIT' split command function but also lose the capability to use 'GETVAR' edit command function. Thus, to change the PF2 key back to the original 'SPLIT' command is not recommended.

Now you may type in some new code in the comment box and use the PF2 key to join the split code line. For example, if you add a word '**NEW**' next to the word '**The**' in line 2 and then press the PF2 key, then a message of '**Spill occurred**' will be prompted on the screen and the result of the merged comment code lines is shown as follows:

Figure 9. The method of joining the code in a comment box

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(XSPLIT)                 Columns 00001 00072      |
|  Command ===> _____________________________________ Scroll ===> CSR           |
|  ****** *************************** Top of Data ***********************      |
|  000001  /*-------------------------------------------------------------*/     |
|  000002  /* XSPLIT - The NEW extended Split and Join function for Edit  */     |
|  000003  /* Macro.                                                      */     |
|  000004  /*-------------------------------------------------------------*/     |
```

```
|  000005  'ISREDIT MACRO (PARM)'                                                |
|  000006  If rc ^= 0 then                                                       |
|  000007     Exit                                                               |
|  000008  If parm = '?' then                                                    |
|  000009     Do                                                                 |
|  000010        Call help                                                       |
|  000011        Exit                                                            |
|  000012     End                                                                |
|  000013  'ISREDIT (CURROW,CURCOL) = CURSOR'                                    |
|  000014                                                                        |
+--------------------------------------------------------------------------------+
```

Note:

1. By adding the comment box in the program source code, the clarity of the program logic can be improved a great deal. In this section, you will find that a 'CMTBOX' Edit Macro command has already been provided by this tool which can help you to insert the comment boxes in your program source very easily.

2. If the cursor is placed on the edit command line or at the prefix line command area when the PF2 key is pressed, then the ISPF screen will be split just like the original ISPF PF2 key command function. The only difference is that the cursor remains on the top edit command line. You will need to press the PF9 (Swap) key to swap the cursor to the new split screen command line, which is a little bit awkward to use.

   Therefore, if you prefer to keep the PF2 key as the original ISPF split key function, you may enter a 'KEYS' command in edit to display a Keylist Utility panel and change the command function on PF2 key entry from 'XSPLIT' to 'SPLIT'.

   In this case, however, you may still keep the PF14 key as the 'XSPLIT' edit command function. To use the PF14 key to split or join the data you will need to press the Shift key and PF2 key at the same time, which is also a little bit awkward to use. It is highly recommended that you may keep the PF2 key as the default 'XSPLIT' edit command because by using this PROJWRK tool you don't really need to split the ISPF screen very often.

B.5 The descriptions of the 'XSWAP' edit command

The XSWAP is an Edit Macro command which can be used for the extended text data cut function, i.e. the 'GETVAR' edit command function, and paste function, i.e. the 'PUTVAR' edit command function. The PF9 key in edit has already been assigned to this new edit command function.

If the cursor is placed on the edit command line area or at the prefix line command area when the PF9 key is pressed, then the original ISPF split screen swap function will be performed. If the cursor is placed at the file area in edit when the PF9 key is pressed, then the 'GETVAR' and 'PUTVAR' functions will be performed depending upon whether the cursor is place at a blank or non−blank area in the edit file area. More detail descriptions of these two edit commands will be described in next section.

B.6 The descriptions of the 'GETVAR' and 'PUTVAR' edit commands

When you develop a program code, have you ever had the trouble with the long variable names because it is very hard to remember them and you have to move the edit screen back to the top of the Variable Declaration Section so many times, which is not only very tedious but also very time consuming?

Sometimes, you may use the 'Cut' and 'Paste' edit commands, which have been described in the previous section, to copy the variable names. However, that method will cut and paste the entire code line and part of the data on the code line may not what you need.

Sometimes, you may use the mouse pointer with the copy and paste buttons to copy and paste the text data on the edit screen. However, that method only allows you to copy and paste one single variable name each time.

The GETVAR and PUTVAR are the two Edit Macro commands that are very similar to the 'Cut' and 'Paste' Edit Macro commands and it is also very much like the mouse pointer with the 'Cut' and 'Paste' buttons method. The 'GETVAR' edit command can be used to copy more than one text data, such as the variable names, the subroutine names, and the label names, etc. on the edit screen and it saves them in a database file. The 'PUTVAR' edit command allows you paste any of the collected text data at any location in the edit file as many times as you like. Thus, this is an extended 'Cut' and 'Paste' edit command function.

The short command forms of these two edit commands are the 'GV' and 'PV' edit commands, respectively. To use these two commands, initially you may enter a 'GV' edit command on the edit command line and position the cursor at any location of a non−blank text data on the edit screen, and then press the Enter key to fetch it.

Instead of using the 'GV' command you may place the cursor at a non−blank area in the edit file and press the PF9 key, i.e. the 'XSWAP' command key, to fetch the edit screen text data.

After the edit screen text data is fetched, it will be saved in the shared buffer. Now you may use a 'PV' edit command to retrieve it from the buffer and paste back to the edit file. To do so, simply enter a 'PV' edit command on the edit command line and position the cursor at any blank space in the file area of the same edit file and then press the Enter key. Instead of using 'PV' edit command you may place the cursor at any blank area and press the PF9 key, i.e. the 'XSWAP' command key, again.

Note:

1. If you use the 'GV' or 'PV' edit command to fetch or paste the screen text data in edit, the cursor must be in the position of the file area of the edit file when the Enter key is pressed.

2. The fetched edit screen text data is available to be pasted to any edited file. You may place the cursor at any blank space in edit and use a 'LISTVAR' or 'LV' edit command to display a 'GVLIST' panel to perform such function. The detail information of the 'LISTVAR' command will be described later.

3. You may enter a 'GV C' or 'PV C' command to clear the fetched data of an edit file from the database as you like. You may enter a 'GV ?' or 'PV ?' command for more detail information about the 'GETVAR' and 'PUTVAR' edit commands.

4. Similar to the 'XCUT' and 'XPASTE' edit commands, the 'GETVAR' and 'PUTVAR' edit commands are also the most useful tools provided by this tool package. For example, the variable names in most of the COBOL programs are very lengthy, and the variable names in many C programs are in the mixed−case format. Those text data are usually very difficult for the programmers to remember while they are writing the program code. When you are assigned to modify a program that was written by someone else and you are not familiar with most of the variable names in the source code, with the usages of these two Edit Macro commands you will find that a lot of your precious time can be saved.

5. If the cursor is at the edit file area when the PF9 key is pressed, and if you don't like it peforms the 'GETVAR' and 'PUTVAR' edit command functions, then you may enter a 'KEYS' command on the edit command line to display a KEYLIST utility panel and change the 'XSWAP' PF key function at PF9 key back to the original 'SWAP' function.

B.7 The examples of the 'GETVAR' and 'PUTVAR' edit commands

The following two diagrams can describe how to use the "GV' and 'PV' edit commands in edit. I personally prefer to use the PF9 key for these two functions. However, in the following sample diagrams the commands to be used for the Copy and Paste of the screen text data functions are in the 'GV' and 'PV' edit command formats for better descriptions.

The following is a part of the source code of a COBOL program. On the edit command line let us enter a 'GV' edit command and place the cursor under any character, say the character 'A', of the variable name "110−GET−TABLE−NAME" at line 3511. After the Enter key is pressed, that variable name will be fetched and saved in a database file.

Figure 10. The method of using the 'GV' edit command to fetch the screen data

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.COBOL(CUSTOMER)               Columns 00001 00072     |
|  Command ===> _gv_____________________________________ Scroll ===> CSR       |
|  ****** *************************** Top of Data ***********************     |
|  003500 *                                                                     |
|  003501  PROCEDURE DIVISION.                                                  |
|  003502                                                                       |
|  003503  000-DELETE-CUSTOMER-ROWS.                                            |
|  003504 *                                                                     |
|  003505      PERFORM 100-PROCESS-CUSTOMER-ROW                                 |
|  003506          UNTIL END-OF-DELETES.                                        |
|  003507      STOP RUN.                                                        |
|  003508 *                                                                     |
|  003509  100-PROCESS-CUSTOMER-ROW.                                            |
|  003510 *                                                                     |
|  003511      PERFORM 110-GET-TABLE-NAME.              <===                    |
|  003512      IF NOT END-OF-DELETES                                            |
|  003513          PERFORM 120-GET-CUSTOMER-NUMBER.                             |
|  003514      IF NOT END-OF-DELETES                                            |
|  003515          PERFORM 130-DELETE-CUSTOMER-ROW                              |
|  003516          PERFORM 140-DISPLAY-STATUS-MESSAGE.                          |
|  003517 *                                                                     |
|  003518                                                                       |
+-------------------------------------------------------------------------------+
```

Instead of using the 'GV' edit command it is highly recommended that you may place the cursor under the character 'A' and press the PF9 key to fetch the variable name "110−GET−TABLE−NAME".

Note that the line 3518 in this edit screen happen to be a blank line. If a 'PV' edit command is entered and the cursor is placed at column 8 on that code line when the Enter key is pressed, then the variable name "110−GET−TABLE−NAME" will be immediately pasted at that location on the code line 3518, which is as shown below:

Figure 11. The method of using the 'PV' edit command to paste the screen data

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.COBOL(CUSTOMER)               Columns 00001 00072     |
|  Command ===> _pv_____________________________________ Scroll ===> CSR       |
|  ****** *************************** Top of Data ***********************     |
|  003500 *                                                                     |
|  003501  PROCEDURE DIVISION.                                                  |
```

```
|  003502                                                                      |
|  003503  000-DELETE-CUSTOMER-ROWS.                                           |
|  003504 *                                                                    |
|  003505      PERFORM 100-PROCESS-CUSTOMER-ROW                                |
|  003506          UNTIL END-OF-DELETES.                                       |
|  003507      STOP RUN.                                                       |
|  003508 *                                                                    |
|  003509  100-PROCESS-CUSTOMER-ROW.                                           |
|  003510 *                                                                    |
|  003511      PERFORM 110-GET-TABLE-NAME.                                     |
|  003512      IF NOT END-OF-DELETES                                           |
|  003513          PERFORM 120-GET-CUSTOMER-NUMBER.                            |
|  003514      IF NOT END-OF-DELETES                                           |
|  003515          PERFORM 130-DELETE-CUSTOMER-ROW                             |
|  003516          PERFORM 140-DISPLAY-STATUS-MESSAGE.                         |
|  003517 *                                                                    |
|  003518  110-GET-TABLE-NAME_                          <===                   |
+------------------------------------------------------------------------------+
```

The cursor is now intelligently shifted next to the new pasted variable name, which allows you to continue to write the code next to it very easily.

Instead of using the 'PV' edit command it is also highly recommended that you may place the cursor at the same location of that code line and press the PF9 key to paste the variable name "110−GET−TABLE−NAME".

If more than one different variable names have been fetched previously on the edit screen before, then whenever the 'PV' edit command is executed in the same edit file the 'GETVAR' selection panel, which will be described in next page, will be displayed for you to select a variable name to be pasted in the edit file.

B.8 The 'GETVAR' selection panel on−line tutorial

The following is an example of the 'GETVAR' selection panel, on which it shows that several screen text data have been fetched by using the 'GV' edit command in edit previously:

Figure 12. The sample GETVAR variable names selection panel

```
+------------------------------------------------------------------------------+
|  GETVAR       Select The Fetched Screen Text Data              Row 1 to 5 of 5|
|  ----------------------------------------------------------------------------|
|  Data Set Name ===> USERID.TEST.COBOL(CUSTOMER)                               |
|  Command ===> _____________________________________________ Scroll ===> CSR  |
|                                                                              |
|   Select  Code  The Previously Fetched Screen Text Data (Paste Mode)         |
|  -------- ---- ------------------------------------------------------------|
|  _________   1   100-PROCESS-CUSTOMER-ROW                                    |
|  _________   2   110-GET-TABLE-NAME                                          |
|  _________   3   120-GET-CUSTOMER-NUMBER                                     |
|  _________   4   130-DELETE-CUSTOMER-ROW                                     |
|  _________   5   140-DISPLAY-STATUS-MESSAGE                                  |
|  ******************************* Bottom of data ********************************|
+------------------------------------------------------------------------------+
```

As you can see that each of these variable names are quite lengthy and not easy to remember. When you write a COBOL program and constantly need to refer the variable names that are defined previously in the other edit screens, then by using this method you can save a lot of precious time.

The screen text data of a program file can be saved in the database for a long period of time until you add or

delete any project or work item.

You may add, delete, update, cut, and paste the data entries on the GETVAR panel as you wish. The detail information of how to perform these functions will be described in detail in next page.

Note that the variable names fetched by the 'GV' edit command do not contain the comma sign. In fact, when you fetch the variable names on the COBOL program edit screen, the arithmetic operator signs, such as '+', '*', '/', the special characters, and the blank space on the edit screen will be converted to white space first, and thus they will not be fetched together with the variable names.

For the non−COBOL program edit screen, all of such characters as well as the '−' sign will not be fetched together with the variable name.

Whenever you use the 'U' command to update the variable name on the panel, you may type any characters, which including the imbedded blank space, the '|' character, and the semicolon. In other words, you may use any characters in the updated variable name field to create new text data entry on the panel.

If you press the PF1 key on this GETVAR selection panel, then the following tutorial screen will be displayed:

Figure 13. The introduction tutorial guide of the 'GETVAR' panel

```
+-------------------------------------------------------------------------------+
|                                                                               |
|                              Introduction                                     |
|                                                                               |
|       The purpose of this function is to allow the users to select the        |
|       previously fetched edit screen text data obtained by using the          |
|       "GETVAR" Edit Macro command and paste it at any location in the         |
|       same edited file by using a "PUTVAR" Edit Macro command. The short      |
|       command forms these two edit commands are the "GV" and the "PV"         |
|       edit commands, respectively.                                            |
|                                                                               |
|       This function is very similar to the "XCUT" and "XPASTE" Edit Macro     |
|       command functions. The "XCUT" and "XPASTE" commands deal with the       |
|       entire code lines but these two commands only deal with the text        |
|       data on the code line. This function is also very similar to the        |
|       function of using the mouse pointer with the "Copy" and "Paste"        |
|       buttons on the edit tool bar. However, the mouse pointer method        |
|       can deal with only one screen text data each time and this method       |
|       can save the fetched text data and paste them multiple times. Thus,     |
|       this method is better than both of the "Cut" and "Paste" command       |
|       method and the mouse pointer method.                                    |
|                                                                               |
+-------------------------------------------------------------------------------+
```

If you press the Enter key or the PF8 key, then the following tutorial guide will be displayed:

```
--------------------------------------------------------------------------------

Display panel      On the edit command line, you may enter a 'GV LIST' or
                   'PV LIST' command to display this GETVAR panel in the List
                   Mode. The short form of these two commands is the 'GVL'
                   command. If more than one screen text data of an edited
                   file have been fetched previously, then whenever you enter
                   a 'PV' command in edit to paste the text data, this GETVAR
                   panel will be displayed in the Paste Mode for you to select
                   one data entry to paste it in edit.
```

Note:	A GVLIST panel can be displayed by using a 'LISTVAR' or 'LV' command on either the edit command line or any panel command line area. On the GVLIST panel, you may enter a selection code to display the GETVAR panel. The GETVAR panel displayed from the GVLIST panel is in the List Mode if the cursor is in the edit command line area. If the cursor is in the edit file area, the the GETVAR panel will be displayed in the Paste mode.
Select an entry	You may either enter a selection code, type a '/' or 'S' command code next to a data entry, or place the cursor at any column of a data entry and press the Enter key to select a data entry. Only one data entry can be selected each time on this GETVAR panel. If the GETVAR panel is displayed in the Paste mode, then the selected data entry will be pasted into the edited file. If the GETVAR panel is displayed in the List mode, then a window panel will be displayed for you to update the selected data entry.
Note:	The data entries on this GETVAR panel were fetched by using the 'GV' command from the edit screen previously. They are saved in the shared pool buffer.
Delete data entry	Type a 'D' command code next to a data entry can delete it from the shared buffer. More than one 'D' command code can be specified to delete multiple data entries at the same time. You may also type a pair of 'DD' command code to delete a block of data entries.
Update data entry	Type an 'U' command code next to a data entry can display an UPDTVAR window panel, on which you may update the text data of the variable name to a new format. Only one 'U' command code to update one variable name field at a time is allowed.
Note:	If this GETVAR panel is displayed in List Mode, whenever a data entry is selected, it will be displayed on an UPDTVAR window panel for you to update instead of being pasted into the edit file. In this case, the 'U' command code can be omitted.
Note:	The variable names on the UPDTVAR window panel will be case−sensitive and you may use any characters, including the imbedded blank space, the '|' character, and the semicolon.
Cut data entry	Type a 'C' command code next to a data entry can cut it from the GETVAR panel into a shared pool buffer. More than one 'C' command code can be specified to cut multiple entries at the same time. You may also type a pair of 'CC' command code to cut a block of data entries.
Paste data entry	Enter a 'XPASTE' or 'P' command on the GETVAR panel command line can paste the previously cut data entries to the GETVAR panel. The cut data entries can either be obtained by the 'C' or 'CC' command on this GETVAR panel or the 'XCUT' edit command executed in the edit file.
Note:	Similarly, the cut data entries obtained from the GETVAR panel can be either pasted to another GETVAR panel or pasted to an edit file.
Add new entry	Enter an 'ADD' command on the GETVAR panel command line can display an ADDVAR window panel to allow you to add new variable names into the GETVAR buffer. The short form of the 'ADD' command is the 'A' command.

Note:	On the edit command line you may enter an 'ADDVAR' command to display the ADDVAR window panel directly without needing to use an 'ADD' command on the GETVAR panel. The short form of the 'ADDVAR' command is the 'AV' command.
Note:	The variable names on the ADDVAR window panel will also be case−sensitive and you may use any characters. Five data entry fields are available on the ADDVAR panel for you to enter the new variable names.
Edit buffer data	Enter an 'EDIT' or an 'E' command on the panel command line can edit the GETVAR temporary work buffer which allows you to add, delete, or update the data entries in the GETVAR shared buffer very easily.
Note:	The 'E' command allows you to add new data entries in the GETVAR panel as well as modifying the code of an existing data entry. Thus, the 'E' command covers the 'U' (Update) command function.
Note:	On the edit command line you may enter an 'EDITVAR' command to edit the GETVAR temporary work buffer directly without needing to use an 'EDIT' command on the GETVAR panel. The short form of the 'EDITVAR' command is the 'EV' command.
Note:	The record length of the GETVAR temporary work buffer is 120 bytes. By using this method, you may use the Cut and Paste commands to save few code lines from the edit file into the shared buffer, and then you may use the 'C' or 'CC' command to cut from the GETVAR panel, and then you may use the 'P' edit command to paste the code lines back to the edit file.
SORT Command	The fetched edit screen text data originally are not sorted. You may enter a 'SORT' command on the panel command line to sort the data entries in ascending order.
RESET Command	To display the sorted data entries back to its original non−sorted order, you may enter a 'RESET', 'RES', or '/R' command.
SAVE Command	After the data entries are sorted by the 'SORT' command, you must enter a 'SAVE' or 'SV' command to save the sorted data entries in the shared buffer. Otherwise, if you just press the PF3 key or enter a 'CAN' command to exit the GETVAR panel, next time when yhou display the GETVAR panel again, the data entries on the panel will still be remained unsorted.
CLEAR Command	On this selection panel you may enter a 'CLEAR' command to clean up the entire fetched text data entries of the selected data set from the GETVAR shared buffer. The short form of the 'CLEAR' command is the 'CL' command. Other than this command, you may also use a 'DELETE' or 'DEL' command to empty the GETVAR shared buffer.
Note:	On the GVLIST panel you may also use a 'CLEAR' command to remove the screen text data from buffer. The differences is that the 'CLEAR' command on GVLIST will delete the GETVAR shared buffer, but the 'CLEAR' command on the GETVAR panel only remove the entire screen text data entries of the selected data set.
Note:	If this GETVAR panel is displayed due to an 'ALL' command is

```
                    entered on the GVLIST panel, then the GETVAR panel is in the
                    'ALL' command mode. In this mode, some of the commands, such
                    as the 'D', 'U', 'E', 'A', 'SORT', 'RES', etc. commands all
                    will not work. The GETVAR panel displayed by using the 'ALL'
                    command on the GVLIST panel is to be used for pasting the
                    screen data only.

------------------ End of GETVAR panel on-line tutorial ----------------------
```

B.9 The 'GVLIST' selection panel on−line tutorial

In the previous section, the GVLIST panel has been mentioned. This panel can be displayed if a 'LISTVAR' or 'LV' command is entered on any process panel command line or the edit command line. The following is an example of the GVLIST selection panel:

Figure 14. The sample GVLIST data set name selection panel

```
+-------------------------------------------------------------------------------+
|  GETVAR       The GETVAR List Utility Process Panel            Row 1 to 4 of 4|
|  -----------------------------------------------------------------------------|
|  Command ===> _____________________________________________ Scroll ===> CSR  |
|   Select  Code  The GETVAR Data Set Name List                                 |
|  -------- ---- ------------------------------------------------------------|
|  ________   1   USERID.TEST.COBOL(CUSTOMER)                                   |
|  ________   2   USERID.TEST.C(HELLO)                                          |
|  ________   3 - USERID.TEST.EXEC(XSPLIT)                                      |
|  ________   4   USERID.TEST.EXEC(HELLO)                                       |
|  ******************************* Bottom of data ********************************|
+-------------------------------------------------------------------------------+
```

On this sample GVLIST panel a PDS member name list of four data sets has been shown. Each data set contains the screen text data which have been fetched by using the 'GV' edit command in edit previously.

Note: On the above sample GVLIST panel the third entry has been flagged with a "−" marker. It means that the 'LV' command is entered on the edit command line of the edited PDS member named 'XSPLIT' of the 'USERID.TEST.EXEC' file. If the 'LV' command is entered on any process panel command line, then the '−' marker will not be shown initially.

This GVLIST panel can be use for maintaining the GETVAR database file. It is a very useful tool of the extended 'Cut' and 'Paste' edit command function.

If you press the PF1 key on this GVLIST selection panel, then the following tutorial screen will be displayed:

Figure 15. The introduction tutorial guide of the 'GVLIST' panel

```
+-------------------------------------------------------------------------------+
|                                                                               |
|                              Introduction                                     |
|                                                                               |
|       This GVLIST panel can be used to display a name list of all the data    |
|       sets of which you have issued the 'GETVAR' command to fetch the edit    |
|       screen data before.                                                     |
|                                                                               |
|       On this GVLIST panel, you may type a 'D' (Delete) command next to       |
|       a data entry to delete it. More than one 'D' command can be specified   |
|       to delete multiple data set names on the panel.                         |
|                                                                               |
```

```
|       You may also enter a selection code or place the cursor at any column    |
|       of a data entry on the panel and press the Enter key to display a       |
|       GETVAR panel of the selected data set.                                  |
|                                                                               |
|       On both this GVLIST panel and the GETVAR panel command line, you may   |
|       enter a 'CL' (Clear) to clean-up all data entries on the panel. Thus,   |
|       this GVLIST panel and the GETVAR panel are very good tools to assist    |
|       you to keep the GETVAR database file in a manageable size.              |
|                                                                               |
+-------------------------------------------------------------------------------+
```

If you press the Enter key or the PF8 key, then the following tutorial guide will be displayed:

Display GVLIST Panel	On any panel command line or edit command line area, you may enter a 'LISTVAR' command to display this GVLIST panel. The short form of the 'LISTVAR' command is the 'LV' command.
Select an entry	On this GVLIST panel you may enter a selection code, place the cursor at any column of a data entry and press the Enter key, or type a 'S' or '/' code next to a data set to display the GETVAR panel of the selected data set. The GETVAR panel displayed will be in the List Mode if the cursor is on the edit command line area after the 'LV' command is entered. If the cursor is placed in the edit file area when the 'LV' command is entered, then the GETVAR panel will be displayed in the Paste Mode.
Note:	The most recently selected entry on this GVLIST panel will always be flagged with a '-' marker. If you want to select the same entry from the panel again, then simply place the cursor on the panel command line area and press the Enter key without entering any command code.
ALL Command	On this GVLIST panel you may enter an 'ALL' command to display a GETVAR panel of the screen text data of all the entries on the GVLIST panel. In this case, the GETVAR panel is displayed in the 'ALL' command mode.
Edit a data set	If you type an 'E' code next to a data set on this GVLIST panel, then the selected data set will be edited. In the edit file, you may use the 'GV' edit command again to fetch the text data from the edit screen.
Note:	Similarly, you may type a 'B' code next to a data set on the GVLIST panel to browse the selected file, or type a 'V' code next to a data set on the GVLIST panel to view the selected file.
CLEAR Command	On this GVLIST panel you may enter a 'CLEAR' command to delete the entire GETVAR database file. The short form of the 'CLEAR' command is the 'CL' command. Other than this command, you may also use a 'DELETE' or 'DEL' command to empty the GETVAR database file.
Note:	On the GETVAR panel you may also use a 'CLEAR' command to remove the screen text data from database. The differences is that the 'CLEAR' command on GVLIST will delete the entire GETVAR database, but the 'CLEAR' command on the GETVAR panel will remove the screen text data entries of the selected data set only.
Note:	By using this GVLIST panel, it allows you to display the GETVAR panel of several different data sets. On the GETVAR

```
                    panels, you may use the 'C' and 'P' command to Copy and
                    Paste the variable names from one GETVAR panel to another.

------------------ End of GVLIST panel on-line tutorial ----------------------
```

B.10 Use 'Cut' and 'Paste' method with the GETVAR panels

In edit file you may use the 'G' (Get) or 'XCUT' Edit Macro command and a pair of the 'CC' line command to cut a block of code lines into the buffer, then you may enter an 'EV' edit command to edit a GETVAR temporary work file and paste the cut code lines into the GETVAR database.

Similarly, on the GVLIST panel you may select several GETVAR panels of different data sets and use the 'C' (Copy) and 'P' (Paste) edit commands to copy the screen text data from one GVLIST data set entry to another. The following is the sample diagrams of how to copy and paste the screen text data among different GVLIST data set entries:

Assume that six text data entries have already been fetched from the 'USERID.TEXT.EXEC(XSPLIT)' data set, which is as shown in a GETVAR panel selected from the GVLIST panel as shown in previous sample diagram. If a pair of 'CC' command code are typed next to the first and the fifth data entry on the panel, then the five selected code lines will be saved in the buffer.

Figure 16. The sample 'CC' command to copy text data from GETVAR panel

```
+-------------------------------------------------------------------------------+
|  GETVAR       Select The Fetched Screen Text Data              Row 1 to 6 of 6|
|  -----------------------------------------------------------------------------|
|  Data Set Name ===> USERID.TEST.EXEC(XSPLIT)                                  |
|  Command ===> _____________________________________________ Scroll ===> CSR  |
|                                                                               |
|   Select  Code  The Previously Fetched Screen Text Data (List Mode)           |
|  -------- ---- ------------------------------------------------------------- |
|  __cc____   1   If parm = '?' length(parm) = 1 then      <===            |    |
|  ________   2      Do                                                         |
|  ________   3         Call help                                               |
|  ________   4         Exit                                                    |
|  ____cc__   5      End                                     <===               |
|  ________   6   'ISREDIT (CURROW,CURCOL) = CURSOR'                            |
|  ****************************** Bottom of data ******************************|
+-------------------------------------------------------------------------------+
```

Now, after pressing the PF3 key the GVLIST panel will be displayed again. On the GVLIST panel, if the 'USERID.TEXT.EXEC(HELLO)' data set entry is selected, then the following sample GETVAR panel will be displayed. Note that when a 'C' or a pair of 'CC' line command is specified on the GETVAR panel, the 'G' or 'XCUT' command is not required. If it is specified, then it will be ignored.

Figure 17. The sample GETVAR panel with a 'P' command

```
+-------------------------------------------------------------------------------+
|  GETVAR       Select The Fetched Screen Text Data              Row 1 to 5 of 5|
|  -----------------------------------------------------------------------------|
|  Data Set Name ===> USERID.TEST.EXEC(HELLO)                                   |
|  Command ===> _p___________________________________________ Scroll ===> CSR  |
|                                                                               |
|   Select  Code  The Previously Fetched Screen Text Data (List Mode)           |
|  -------- ---- ------------------------------------------------------------- |
|  ________   1   If parm = 'SAY' then                                          |
|  ________   2      Do                                                         |
```

```
|  _________   3         Say "Goodbye old world"                                |
|  _________   4         Say "Hellow NEW world"                                 |
|  _________   5      End                                                       |
|  ****************************** Bottom of data ******************************|
+-------------------------------------------------------------------------------+
```

Suppose a 'P' (Paste) command is entered on the GETVAR panel command line, then after pressing the Enter key, the previously cut five code lines from the 'USERID.TEST.EXEC(XSPLIT)' data set will be appended to the bottom of this GETVAR panel, which is as shown below. Note that when a 'P' or 'XPASTE' command is specified on the GETVAR panel, the 'A' or 'B' line command is not required. If it is specified, then it will be ignored.

Figure 18. The sample GETVAR panel with the five pasted code lines

```
+-------------------------------------------------------------------------------+
|  GETVAR       Select The Fetched Screen Text Data             Row 1 to 10 of 10|
|  -----------------------------------------------------------------------------|
|  Data Set Name ===> USERID.TEST.EXEC(HELLO)                                   |
|  Command ===> _____________________________________________ Scroll ===> CSR  |
|                                                                               |
|   Select  Code  The Previously Fetched Screen Text Data (List Mode)           |
|  -------- ---- ------------------------------------------------------------|
|  _________   1   If parm = 'SAY' then                                         |
|  _________   2      Do                                                        |
|  _________   3         Say "Goodbye old world"                                |
|  _________   4         Say "Hellow NEW world"                                 |
|  _________   5      End                                                       |
|  _________   6   If parm = '?' length(parm) = 1 then      <===             |
|  _________   7      Do                                       <===             |
|  _________   8         Call help                             <===             |
|  _________   9         Exit                                  <===             |
|  _________  10      End                                      <===             |
|  ****************************** Bottom of data ******************************|
+-------------------------------------------------------------------------------+
```

By using this method, you may save the important code lines of an edited file in the GETVAR database and paste them back to the edit file very easily at a later time.

Note that instead of pasting the code lines directly to the GETVAR panel, you may either use an 'E' (Edit) command on the GETVAR panel to edit a GETVAR database temporary work file and then paste them in edit. The editing temporary file method will truncate the code line if it is longer than 120 characters. The pasting to GETVAR panel method will preserve the code line no matter how long it is. Note also that the editing temporary file method allows you to paste the code lines at any location in the edit file. However, the pasting to GETVAR panel method can only allow you to paste the code lines to the bottom of the panel.

B.11 How to save code lines with the ADDVAR panel

First of all, you may type a 'G' (Get) line command or a pair of 'GG' line command to cut the code lines from a program file in edit, such as the 'GG' line command pair typed in at line 8 and line 12 as shown in the following diagram:

Figure 19. The method of joining the code in a comment box

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(XSPLIT)                 Columns 00001 00072      |
|  Command ===> ___________________________________________ Scroll ===> CSR      |
|  ****** ***************************** Top of Data ***********************     |
```

```
|  000001  /*-------------------------------------------------------------*/     |
|  000002  /* XSPLIT - The NEW extended Split and Join function for Edit  */     |
|  000003  /* Macro.                                                      */     |
|  000004  /*-------------------------------------------------------------*/     |
|  000005  'ISREDIT MACRO (PARM)'                                               |
|  000006  If rc ^= 0 then                                                      |
|  000007     Exit                                                              |
|  gg0008  If parm = '?' then     <===                                          |
|  000009     Do                                                                |
|  000010        Call help                                                      |
|  000011        Exit                                                           |
|  00gg12     End                 <===                                          |
|  000013  'ISREDIT (CURROW,CURCOL) = CURSOR'                                   |
|  000014                                                                       |
+-------------------------------------------------------------------------------+
```

After few code lines have been cut and saved in a shared pool, you may enter an 'AV' edit command to display an ADDVAR panel, on which you may press the PF11 key to save the cut code lines for the future use. The cut code lines will be listed on the ADDVAR panel when the PF11 key is pressed, which is as shown below:

Figure 20. The sample ADDVAR panel with few cut code lines

```
+-------------------------------------------------------------------------------+
|                                                                               |
|   Enter the New variable name(s) or cut code lines:                           |
|                                                                               |
|  >  If parm = '?' then                                                        |
|  >     Do                                                                     |
|  >        Call help                                                           |
|  >        Exit                                                                |
|  >     End                                                                    |
|  >                                                                            |
|  >                                                                            |
|  >                                                                            |
|  >                                                                            |
|  >                                                                            |
|  >                                                                            |
|  >                                                                            |
|                                                                               |
|   Press ENTER to confirm add function.                                        |
|   Press PF11 to save the cut code lines if any.                               |
|   Press END or CANCEL to cancel add function.                                 |
+-------------------------------------------------------------------------------+
```

On this ADDVAR panel you may either type in any data, or press the PF11 key to list the cut code lines if there is any. After the data line fields are filled with code, you may then press the Enter key to save them in a GETVAR buffer.

Note that this ADDVAR panel initially contains only 12 empty data fields. If you need to save more than 12 code lines in a program, then you need to cut them more than once in order to save them all in the buffer.

In the future whenever you need to paste the previously saved cut code lines, all you need to do is enter a 'LV' edit command to display the GVLIST panel and select a PDS member name to display a GETVAR panel. Then apply the method of using a 'C' command or a pair of 'CC' command, which has been described in previous section, to fetch the data lines from the GETVAR panel. Finally, use a 'P' or 'PK' line command or a 'P' or 'P K' edit command to paste the code lines in the program.

This method is very much better than the 'Cut' and 'Paste' method. By using this method, you may save the cut

code lines in a GETVAR buffer for a long period of time before pasting them in the program.

B.12 The descriptions of the 'COLSHFT' edit command

The COLSHFT is an Edit Macro command which can be used for the extended Column Shift function in edit file. The short form of the 'COLSHFT' edit command is the 'CS' edit command.

Traditionally when you shift the code to the left or right, you need to use one of the '(', '((', ')', '))', '<',' <<', '>', and '>>' commands with a number, which is very tedious. By using the 'CS' edit command you may place the cursor in the file area to determine whether you want the code to be shifted to the left or right and this tool will automatically calculate how many columns you want the code to be shifted.

B.13 The examples of the 'COLSHFT' edit command

The following few diagrams can describe how to use the "COLSHFT" edit command to shift the code in the edit file. Assume a 'CS' edit command is entered at the command line and a pair of 'CC' line commands are typed in on line 3 and line 7, and assume the cursor is placed at column 8 that happen to be under the last character 'm' of the word 'parm', which is as shown below:

Figure 21. The method of entering a 'CS' edit command in edit file

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(HELLO)                  Columns 00001 00072      |
|  Command ===> _cs_____________________________________ Scroll ===> CSR        |
|  ****** **************************** Top of Data ***********************     |
|  000001  /* REXX */                                                           |
|  000002  Arg parm .                                                           |
|  0cc003  If parm = '?' then         <===                                      |
|  000004     Do                                                                |
|  000005        Call help                                                      |
|  000006        Exit                                                           |
|  000cc7     End                     <===                                      |
|  000008  If parm = 'SAY' then                                                 |
|  000009     Do                                                                |
|  000010        Say "Goodbye old world"                                        |
|  000011        Say "Hellow NEW world"                                         |
|  000012     End                                                               |
|  000013                                                                       |
+-------------------------------------------------------------------------------+
```

After pressing the Enter key, the code lines from line 3 to line 7 will be shifted to the right, and the leftmost non−blank character on line 3 will be aligned with column 8, which is shown as follows:

Figure 22. The result of the 'CS' edit command shifted code listing

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.EXEC(HELLO)                  Columns 00001 00072      |
|  Command ===> _____________________________________ Scroll ===> CSR            |
|  ****** **************************** Top of Data ***********************     |
|  000001  /* REXX */                                                           |
|  000002  Arg parm .                                                           |
|  000003        If parm = '?' then                                             |
|  000004           Do                                                          |
|  000005              Call help                                                |
|  000006              Exit                                                     |
|  000007           End                                                         |
```

```
|  000008  If parm = 'SAY' then                                                |
|  000009     Do                                                               |
|  000010        Say "Goodbye old world"                                       |
|  000011        Say "Hellow NEW world"                                        |
|  000012     End                                                              |
|  000013                                                                      |
+------------------------------------------------------------------------------+
```

Note:

1. The above example shows that the cursor on column 8 is to the right of the leftmost non−blank character on line 3, which is at column 2. Thus, this tool determines that you want to shift the block of code lines from line 3 to line 7 to the right for a total of 6 columns.

   In this simple example, you can use a pair of '))' and '))6' line command to get the same result. However, if you want to shift a block of code lines to many columns to the left or right, then the 'CS' edit command method turns out to be a lot easier to use because counting how many columns to be shifted to the left or right is a tedious process.

2. If any code line does not have enough room for shifting the code to the right, then some of the data will be spill over into next code line. Unlike the 'CS' edit command, the ISPF ')' and '))' line commands will truncate the data from the end if there is not enough room in the code line.

3. Similarly, if the cursor position is to the left of the leftmost non−blank character of a block of code lines, then this tool determines that you want to shift the block of code lines to the left.

4. If any code line in the block is the comment line in a comment box, then the 'CS' edit command will be aborted. Please see the 'CMTBOX' edit command, which will be described later in this section, for the information of how to create a comment box in edit.

5. To shift a single line to the left or right using the 'CS' edit command, the 'C' line command can be omitted because the selected code line to be shifted is the cursor line.

Please enter a 'CS ?' command for more detail information about this command function in edit.

B.14 The descriptions of the 'SRCH' and 'XS' edit commands

Just like the 'SRCH' command on the PANEL4 panel and the 'XS' (or 'XSRCH') command on the PANEL3 panel, the 'SRCH' and 'XS' (or 'XSRCH') commands can also be invoked on the edit command line area when the edited file is a PDS member. When either of these two edit commands is invoked in edit, a 'String Search−For Utility' or 'Extended String Search−For Utility' panel will be displayed. You may either enter the searched data string on the panel, or position the cursor in the edit file screen to fetch the data string.

For the 'SRCH' edit command the fetched data string will be converted to uppercase, for the 'XS' edit command the fetched data string will be kept as is and placed on the panel.

Note: If the edited file is a sequential file, then the 'SRCH' and 'XS' edit commands will be automatically converted to the 'FFIND' edit command for the data string search function in the edit file.

The following is an example of how to use the 'XS' edit command in an edit file. Assume that an 'XS' command is entered on the edit command line and the cursor is placed under the character 'r' of the variable 'printf' at line 5, which is as shown below:

Figure 23. The 'XS' edit command with the cursor placed in the file area

```
+------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                      Columns 00001 00072    |
|  Command ===> xs_____________________________________ Scroll ===> CSR        |
|  ****** **************************** Top of Data ************************     |
|  000001  /* Hello.c - sample program */                                       |
|  000002  #include <stdio.h>                                                   |
|  000003                                                                      |
|  000004  void main() {                                                       |
|  000005     printf("Goodbye old world\n");      <===                        |
|  000006     printf("Hello NEW world\n");                                      |
|  000008  }                                                                   |
+------------------------------------------------------------------------------+
```

Then after pressing the Enter key, the following sample 'Extended String Search−For Utility' panel will be shown:

Figure 24. The 'Extended Search−For' penal with a fetched data string 'printf'

```
+------------------------------------------------------------------------------+
|  XSEARCH      Extended Search-For Utility      Menu  Utilities  Options  Help  |
|  ----------------------------------------------------------------------------|
|  Command ===> ____________________________________________________________   |
|                                                                              |
|  Enter Search Strings and Optional operands (WORD/PREFIX/SUFFIX,C)           |
|     Caps . . ____________________________________________________________    |
|     Caps . . ____________________________________________________________    |
|     Caps . . ____________________________________________________________    |
|     Caps . . ____________________________________________________________    |
|     Caps . . ____________________________________________________________    |
|     Asis . . printf______________________________________________________    |
|     Asis . . ____________________________________________________________    |
|     Asis . . ____________________________________________________________    |
|     Asis . . ____________________________________________________________    |
|     Asis . . ____________________________________________________________    |
|                                                                              |
|  Listing Data Set . . SRCHFOR.LIST_______________________________________    |
|  Process Options. . . ___________________________________________________    |
|  Statements Dsn . . . ___________________________________________________    |
|                                                                              |
|  Execution Mode               Output Mode             Edit/View Member       |
|  1  1. Foreground             1  1. View              2  1. Edit             |
|     2. Batch                     2. Browse               2. View             |
+------------------------------------------------------------------------------+
```

Note that on the above diagram only the variable name 'printf' will be fetched. The left parenthesis '(' on the edit screen will be converted to blank space when the variable name is fetched. Thus, the data string to the right of the left parenthesis '(' will not be fetched together with the variable name 'printf'. You may change this data string on the panel as you wish. If the fetched data string is in uppercase, then it will be place in 'Caps' field on the panel instead.

After pressing the Enter key on the above 'Extended String Search−For Utility' panel, the data string of 'printf' in all the member files of the 'USERID.TEST.C' library will be searched and displayed.

This is a very good tool if you wish to find all the same data strings in other members of the same PDS if you find a data string in the edit file that you are interested in.

Note: On the VIEW listing of the SRCHFOR.LIST file, you may enter a 'CPY' or 'FM' edit command to copy all members to a newly created PDS file or displayed in a PDS Member List panel. The valid option codes of the 'FM' command are: 'IN', 'OUT', 'INLIST', and 'OUTLIST', where the option code 'IN' is the default and it

can be omitted.

B.15 The 'AUTOINS' edit command with the Balance Code function

The AUTOINS is an Edit Macro command which can be used for the input data automatic insertion function in edit file. It also contains the Balance Code automatic insertion function and the Intelligent Cursor Movement function. The short command form of the 'AUTOINS' edit command is the 'I' edit command. The PF4 key in edit has already been assigned to this command function. You may also use an 'AI' line command in edit to enter the 'Automatic Insertion' mode.

On the edit command line, if an 'I OPT' or 'IO' command is entered, then the following sample AUTOINS option selection panel will be displayed:

Figure 25. The sample AUTOINS edit command option selection panel

```
+------------------------------------------------------------------------------+
|                                                                              |
|     The Edit Automatic Data Insertion Function Options:                      |
|                                                                              |
|     1) Balance Code Profile Group       (1 to 9) -  1   (Default = 1 )       |
|                                                                              |
|     2) Source code left Zone Margin     (1 to 3) -  1   (Default = 1 )       |
|     3) Source code right Zone Margin  (65 to 72) -  71  (Default = 71)       |
|     4) Amount of cursor movement        (1 to 3) -  3   (Default = 3 )       |
|                                                                              |
|                                                                              |
|     Press PF01 for tutorial guide.                                           |
|     Press ENTER to display Balance Code Profile Group.                       |
|     Press END to return to program source file.                              |
|     Press CANCEL to cancel the option changes.                               |
|                                                                              |
+------------------------------------------------------------------------------+
```

On this option selection panel, you may change the value of some options for your own needs.

B.16 The 'AUTOINS' option selection panel on−line tutorial

If you press the PF1 key on the AUTOINS option selection panel, then the following tutorial screen will be displayed:

Figure 26. The introduction tutorial guide of the 'I' Edit Macro command

```
+------------------------------------------------------------------------------+
|                                                                              |
|                                                                              |
|                              Introduction                                    |
|                                                                              |
|     The 'I' command can be used to insert the code in the Edit file. This    |
|     command can also insert the Balance Codes for the input keyword. For     |
|     example, if you insert a "Do" keyword code in the edit file using this   |
|     command, then it will automatically insert an "End" keyword code in      |
|     the Edit file to balance it. After the Balanced Code is inserted, the    |
|     cursor will be placed in a correct position on the middle code line      |
|     which allows you to continue to enter the data very easily.              |
|                                                                              |
```

```
|     There are various types of Balance Code can be selected for the Balance    |
|     Code Automatic Insertion function. You may enter an "I OPT" command to    |
|     display this panel and pick a valid Balance Code profile group number     |
|     to display the Balance Code option selection panel.                       |
|                                                                               |
|                                                                               |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

If you press the Enter key or the PF8 key, then the following tutorial guide will be displayed:

```
-------------------------------------------------------------------------------

Note:               To display this 'I' command options panel, you need to enter
                    a 'I OPTION' or 'I O' command on the command line of the
                    edited file. The short command form of the 'I O' command is
                    'IO' command. You may enter a 'I ?' command for the detail
                    information of the usages of this command.

Profile Group       There are 9 different Balance Code profile panels available
                    for you to choose. You may enter the valid selection number
                    ranging from 1 to 9 to select. After it is specified, the
                    Balance Code profile panel, i.e. the Balance Code options
                    selection panel, will be displayed. The default is to select
                    the first Balance Code profile. Initially all the options of
                    the Balance Code will be chosen with the '/' code as the
                    default. You may enter a blank at any row to remove the
                    options that you don't want the Balance Code to be inserted.

Left Zone Margin    The column number of the leftmost column of the File Zone of
                    the edited file. The valid range of the left margin is from
                    column 1 to column 3. The default left margin is column 1.

Right Zone Margin   The column number of the rightmost column of the File Zone
                    of the edited file. The valid range of the right margin is
                    from column 65 to column 72. The default right margin is
                    column 71.

Cursor Movement     This is the total amount of the columns of the 'Intelligent
                    Cursor Movement' function. The valid number is from 1 to 3.
                    The default is 3 columns. For example, when the "End" code
                    is inserted to balance the "Do" code, the cursor will be
                    shifted 3 columns to the right in between the "Do" and "End"
                    code lines in order for you to insert a new input line very
                    easily. You may change this amount to either 1 column or 2
                    columns as you wish.

------------------ End of 'I' edit command on-line tutorial ------------------
```

B.17 The 'AUTOINS Balance Code' options selection

When a Balance Code Profile group number is selected on the AUTOINS option selection panel and the Enter key is pressed, the following Balance Code option selection code will be displayed. Initially all the Balance Code options will be selected as default. You may enter the blank on any option to remove it if you don't want that Balance Code option to be performed.

Figure 27. The AUTOINS Balance Code option selection panel

```
+-------------------------------------------------------------------------------+
|                                                                               |
```

```
|                       AUTOINS Balance Code Select Options                    |
|                                                                              |
|   Enter "/" to select option or "blank" to remove.  Scroll to view selection |
|   Press Enter/End to process or Return/Cancel to exit.                       |
|                                                                              |
|                                                                              |
|                       Control Keyword Balance Options                        |
|   /  DO       - Balance the "DO" keyword with the "END" keyword              |
|   /  BEGIN    - Balance the "BEGIN" keyword with the "END" keyword           |
|   /  SELECT   - Balance the "SELECT" keyword with the "END" keyword          |
|   /  CASE     - Balance the "CASE" keyword with the "END" keyword            |
|   /  PROC     - Balance the "PROC" keyword with the "END" keyword            |
|   /  PRCEDUR  - Balance the "PROCEDURE" keyword with the "END" keyword       |
|   /  MACRO    - Balance the "MACRO" keyword with the "END" keyword           |
|   /  LOGIC    - Balance the "@LOGIC" keyword with the "@ENDLOGIC" keyword    |
|                                                                              |
|                       Comment Token Balance Options                          |
|   /  COMMENT  - Balance the "/*" code with the "*/" code                     |
|   /  CMTEXCL  - Balance the "/*!" code with the "*/" code                    |
|   /  CMTDOT   - Balance the "(*" code with the "*)" code                     |
|   /  PARNDOT  - Balance the "(." code with the ".)" code                     |
|   /  BRACE    - Balance the "{" code with the "}" code          <==          |
|   /  BRACKET  - Balance the "[" code with the "]" code                       |
|   /  PAREN    - Balance the "(" code with the ")" code                       |
|                                                                              |
|                       Conditional Keyword Balance Options                    |
|   /  IFTHEN   - Balance the "IF" keyword with the "THEN" keyword             |
|   /  WHENTHN  - Balance the "WHEN" keyword with the "THEN" keyword           |
|   /  RPTUNTL  - Balance the "REPEAT" keyword with the "UNTIL" keyword        |
|   /  FORDO    - Balance the "FOR" keyword with the "DO" keyword              |
|   /  WITHDO   - Balance the "WITH" keyword with the "DO" keyword             |
|   /  WHILEDO  - Balance the "WHILE" keyword with the "DO" keyword            |
|                                                                              |
+------------------------------------------------------------------------------+
```

Note: In the above sample diagram, the balance code '**}**' for the token '**{**' has been selected in the table, which will be used in an example of the C program that will be described later in the section.

B.18 The 'Balance Code' option selection panel on−line tutorial

If you press the PF1 key on the Balance Code option selection panel, then the following tutorial screen will be displayed:

Figure 28. The introduction tutorial guide of the Balance Code options

```
+------------------------------------------------------------------------------+
|                                                                              |
|                              Introduction                                    |
|                                                                              |
|     This panel can be used to define the Balance Code options. On this       |
|     panel the Balance Codes for the PL/I, PASCAL, C, C++, REXX, and PL/X,    |
|     etc. programming languages are all mixed up in the same profile.         |
|     Initially all of the options on the panel are selected by default        |
|     which have marked with the '/' code. If you want to remove some of       |
|     the options from the Balance Code profile, then you may enter a blank    |
|     on the selection code field of those options.                            |
|                                                                              |
|     There are totally 9 Balance Code profile groups available. You may       |
|     enter a valid group number on the Edit Code Automatic Insertion panel,   |
|     i.e. the AUTOINS panel, to display this panel and define your Balance    |
|     Code options. Usually you just need the first Balance Code profile       |
```

```
|     panel for the AUTOINS function. The other 8 Balance Code profiles are      |
|     the auxiliary profiles if you don't like to frequently update the          |
|     Balance Code options on a single profile panel.                            |
|                                                                                |
|                                                                                |
+--------------------------------------------------------------------------------+
```

B.19 The examples of the 'I' edit command

The following few diagrams can describe how the "BALANCE" function works in the AUTOINS mode. Assume an 'I 3' edit command is entered at the command line or an 'I' edit command is entered at the command line with an 'A' line command typed at the line 3 in a sample C program which is as shown below:

Figure 29. The method of entering an 'i 3' edit command in edit file

```
+--------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                        Columns 00001 00072    |
|  Command ===> _i 3_____________________________________ Scroll ===> CSR        |
|  ****** **************************** Top of Data ***********************      |
|  000001  /* Hello.c - sample program */                                        |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                        |
|  000004                                                                        |
+--------------------------------------------------------------------------------+
```

After pressing the Enter key, a new empty code line 4 will be inserted under the line 3, which is labeled with the '.I' flag and the command line will contain an 'AUTOINS' command code. This will bring the 'I' edit command function into the 'Automatic Insertion Mode'.

Note: Instead of using the 'I' or 'I 3' edit command, you may position the cursor at a proper column on line 3 and press the PF4 key to enter into the 'Automatic Insertion Mode' also.

Assume that the 'void main() {' code statement is typed over the new line 4, which is shown in the following diagram:

Figure 30. The new code is typed over the new line 4

```
+--------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                        Columns 00001 00072    |
|  Command ===> AUTOINS__________________________________ Scroll ===> CSR        |
|  ****** **************************** Top of Data ***********************      |
|  000001  /* Hello.c - sample program */                                        |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                        |
|  .I      void main() {                                                         |
|  000005                                                                        |
+--------------------------------------------------------------------------------+
```

Then after you press the Enter key, the new line 5 will be inserted below the line 4 and a code line 6 with the balance code '}' automatically inserted below the new line 5 which will be shown in the following diagram.

Note that a label with the '.I' flag and a command code 'AUTOINS' have already been inserted into the prefix line command area and the edit command line area to maintain the 'Automatic Insertion Mode', and the cursor has been intelligently shifted 3 columns to the right from the leftmost column of the 'main()' statement which allows you to continue to enter other program source code at the right position very easily.

Figure 31. A balance code line 6 is added below the new line 5

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                      Columns 00001 00072     |
|  Command ===> AUTOINS_____________________________________ Scroll ===> CSR     |
|  ****** **************************** Top of Data ***********************     |
|  000001  /* Hello.c - sample program */                                       |
|  000002  #include <stdio.h>                                                   |
|  000003                                                                       |
|  000004  void main() {                                                        |
|  .I         _                                                                 |
|  000006  }                                                                    |
+-------------------------------------------------------------------------------+
```

Now, you may continue to type the code lines over the new empty input code lines, which is shown in the following diagram:

Figure 32. Continue to type the input data over the new lines

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                      Columns 00001 00072     |
|  Command ===> AUTOINS_____________________________________ Scroll ===> CSR     |
|  ****** **************************** Top of Data ***********************     |
|  000001  /* Hello.c - sample program */                                       |
|  000002  #include <stdio.h>                                                   |
|  000003                                                                       |
|  000004  void main() {                                                        |
|  000005     printf("Goodbye old world\n");                                    |
|  000006     printf("Hello NEW world\n");                                      |
|  .I         _                                                                 |
|  000008  }                                                                    |
+-------------------------------------------------------------------------------+
```

While the new line, i.e. the line 7, is still empty and the Enter key is pressed, then the new line will be dropped and the AUTOINS mode will be terminated.

Instead of pressing the Enter key over an empty input code line to stop the AUTOINS mode, you may add an 'END' parameter to the 'AUTOINS' edit command, i.e. enter an 'AUTOINS END' command, to terminate the AUTOINS mode also.

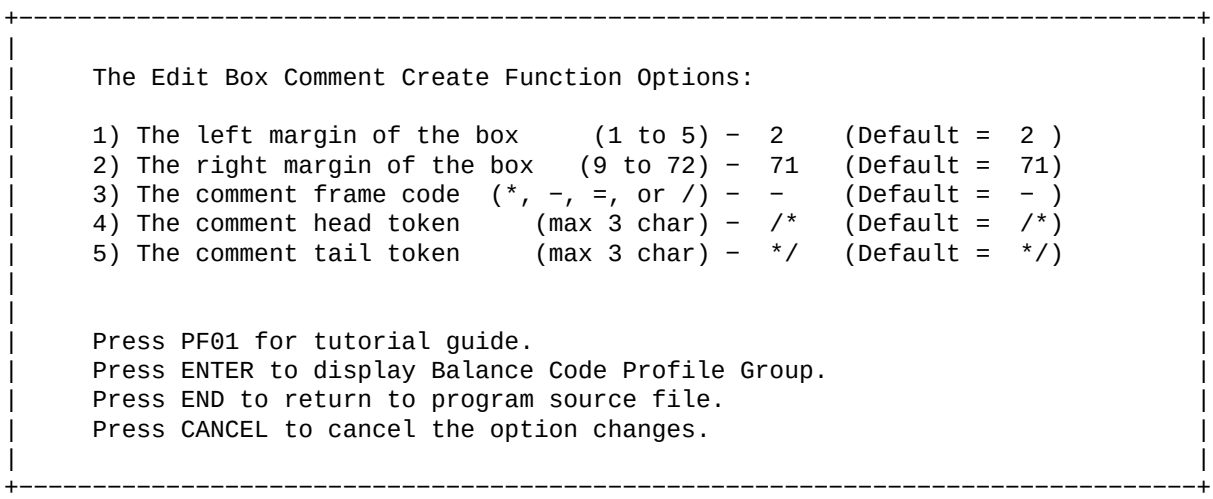
Note: If the edited file is a JCL file, when you place the cursor on a code line and press PF4 key, i.e. the 'ED' command key, if the code line contains a 'DSN=' keyword, then this tool will automatically edit the file specified in the 'DSN=' code. Otherwise it will insert a blank JCL code line with '//' code at the leftmost two columns.

B.20 The descriptions of the 'CMTBOX' edit command

The CMTBOX is an Edit Macro command which can be used for inserting a comment box in edit file. It is a very useful tool for the MVS programmers to document their program code in a very clear and very neat fashion. The short command form of the 'CMTBOX' edit command is the 'BX' edit command. You may also type a 'BX' line command to create a comment box in edit.

On the edit command line, if an 'BX OPT' or 'BXO' command is entered, then the following sample CMTBOX option selection panel will be displayed:

Figure 33. The sample CMTBOX edit command option selection panel

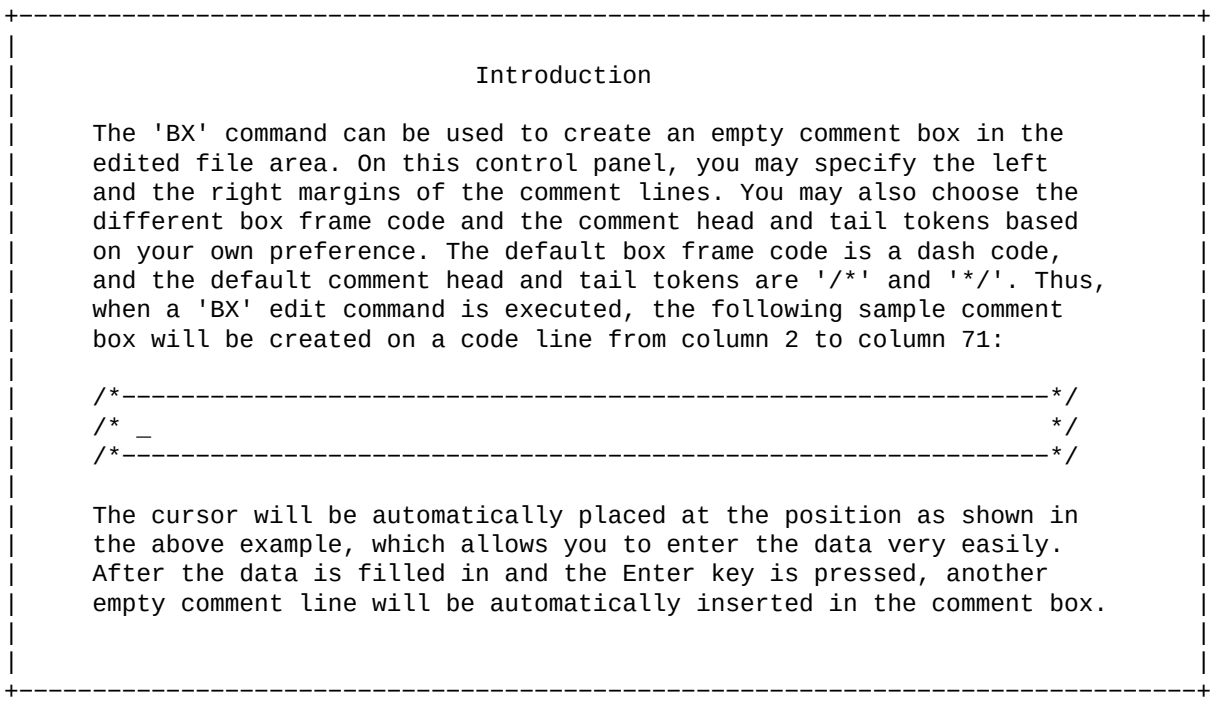


On this option selection panel, you may change the value of some options for your own needs.

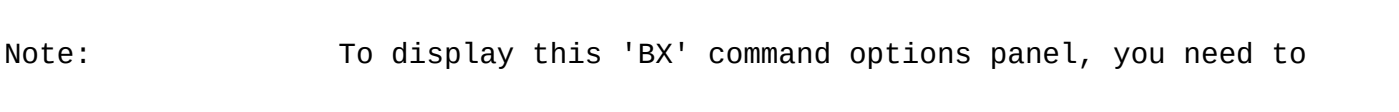
B.21 The 'CMTBOX' option selection panel on-line tutorial

If you press the PF1 key on the CMTBOX option selection panel, then the following tutorial screen will be displayed:

Figure 34. The introduction tutorial guide of the 'BX' edit command



If you press the Enter key or the PF8 key, then the following tutorial guide will be displayed:



	enter a 'BX OPTION' or 'BX O' command on the command line of the edited file. The short form of the 'BX O' command is the 'BXO' command. You may enter a 'BX ?' command for the detail information of the usages of this command.
Left Margin	The column number of the leftmost column of the comment box. The valid range of the left margin is from column 1 to column 5. The default left margin is column 2. If you want to create a comment box start from a column that is beyond column 5, then simply enter a 'BX' command and then place the cursor at that column in the file area, and then press the Enter key to create a smaller comment box.
Right Margin	The column number of the rightmost column of the comment box. The valid range of the right margin is from column 9 to column 72. The default right margin is column 71.
Box Frame Code	The box frame code can be used to build the boundary line of the comment box. The valid frame codes are '*', '-', and '=' and default frame code is '-'. For C++ programs, you may also specify a '/' as the box frame code.
Comment Head	The box comment head is the left comment delimiter of the comment box. The valid comment head must be less than 3 characters. The default comment head token is '/*'. The other commonly selected comment head tokens are '//', '/*!', '/**', and '{', etc.
Comment Tail	The box comment tail is the right comment delimiter of the comment box. The valid comment head must be less than 3 characters. The default comment head token is '*/'. The other commonly selected comment tail tokens are '//', '**/', and '}', etc. If you want to set the comment tail token to blank, then you must enter a 'NUL' keyword in this field.

---------- End of 'BX' edit command on-line tutorial ----------

B.22 The examples of the 'BX' edit command

The following few diagrams can describe how the 'CMTBOX' edit command function works in the edit file. Assume a 'BX 3' edit command is entered at the command line or a 'BX' edit command is entered at the command line with an 'A' prefix line command typed at line 3 in the sample C program which is as shown below:

Figure 35. The method of entering a 'bx 3' edit command in edit file

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                      Columns 00001 00072     |
|  Command ===> _bx 3_____________________________________ Scroll ===> CSR       |
|  ****** *************************** Top of Data ***********************      |
|  000001  /* Hello.c - sample program */                                        |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                       |
|  000004  void main() {                                                        |
|  000005     printf("Goodbye old world\n");                                    |
|  000006     printf("Hello NEW world\n");                                      |
|  000007  }                                                                    |
+-------------------------------------------------------------------------------+
```

After pressing the Enter key, a new comment box will be created and inserted below the line 3, which is shown in the following diagram. Note that the cursor has been intelligently placed at a column position on the

middle empty comment line which has labeled with a '.BX' flag. The command line will contain a 'CMTBOX' edit command code. This will bring the 'BX' edit command function into the 'Automatic Insertion Mode' within the CMTBOX mode.

Figure 36. A new comment box created below the line 3

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                    Columns 00001 00072       |
|  Command ===> CMTBOX_____________________________________ Scroll ===> CSR      |
|  ****** **************************** Top of Data ***********************      |
|  000001  /* Hello.c - sample program */                                        |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                       |
|  000004  /*-----------------------------------------------------------------*/ |
|  .BX     /* _                                                              */ |
|  000006  /*-----------------------------------------------------------------*/ |
|  000007  void main() {                                                        |
|  000008     printf("Goodbye old world\n");                                    |
|  000009     printf("Hello NEW world\n");                                      |
|  000010  }                                                                    |
+-------------------------------------------------------------------------------+
```

If any input data is typed over the empty comment line area and the Enter key is pressed, then a new empty comment line will be inserted below the line 5 and the cursor will be intelligently place at a position which allows you to continue to enter the text data in the next empty comment box code line in an Automatic Insertion Mode. The following diagram shows how it works:

Figure 37. A new empty comment box line inserted below line 5

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                    Columns 00001 00072       |
|  Command ===> CMTBOX_____________________________________ Scroll ===> CSR      |
|  ****** **************************** Top of Data ***********************      |
|  000001  /* Hello.c - sample program */                                        |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                       |
|  000004  /*-----------------------------------------------------------------*/ |
|  000005  /* This is a sample C program named Hello.c, and this box is a */    |
|  .BX     /* _                                                              */ |
|  000007  /*-----------------------------------------------------------------*/ |
|  000008  void main() {                                                        |
|  000009     printf("Goodbye old world\n");                                    |
|  000010     printf("Hello NEW world\n");                                      |
|  000011  }                                                                    |
+-------------------------------------------------------------------------------+
```

If no input data is typed over the empty comment line when the Enter key is pressed, then the entire new comment box or the new added empty comment box code line will be dropped and the CMTBOX mode will be terminated. You may create an empty comment box by entering two dots '..' on the input comment line before pressing the Enter key.

Instead of pressing the Enter key over an empty comment line to stop the CMTBOX mode, you may add an 'END' parameter to the 'CMTBOX' edit command, i.e. enter an 'CMTBOX END' edit command, to terminate the CMTBOX mode also.

If you enter a 'BX' edit command and place the cursor in the middle of a comment box code line, then that code line can be either split into two comment box code lines or join the next comment box code line to

construct a single comment box code line.

The default comment frame code is a dash '−'. You may specify a comment frame code asterisk '*', dash '−', equal sign '=', or slash '/' as an option code in the 'BX' edit command to temporarily alter the output comment box frame format. For example, the 'BX *', 'BX =', 'BX −', and 'BX /' are all valid edit commands. In this case, the original frame code defined on the 'BX' Option Panel will be remained unchanged.

Similarly when the 'BX' edit command is entered on the edit file command line, instead of specifying a line command code 'A' or 'B' and keep the cursor in the prefix line command area, you may position the cursor in the file area to temporarily enlarge the left margin value and create a smaller comment box. In this case, the original value of the left margin defined on the 'BX' Option Panel will be remained unchanged.

If you enter a 'BX F' edit command and place the cursor on the prefix line command area of a comment box code line either with or without specifying a line command code 'A', then the comment box code lines will be formatted. The short form of the 'BX F' edit command is the 'BXF' edit command. You may enter a 'BXF num' edit command to format 'num' of comment box lines counting from the cursor line. For example, the 'BXF 2' edit command can be used to format 2 comment box code lines below and include the cursor line.

B.23 The descriptions of the 'CMTLINE' edit command

The CMTLINE is an Edit Macro command which can be used for inserting an empty remark comment line in edit file. It is a very useful tool for the MVS programmers to document their source code in a very neat format. The short command form of the 'CMTLINE' edit command is the 'CL' edit command. You may also type a 'CL' line command to create an empty remark comment line in edit.

On the edit command line, if an 'CL OPT' or 'CLO' command is entered, then the following sample CMTLINE option selection panel will be displayed:

Figure 38. The sample CMTLINE edit command option selection panel

```
+------------------------------------------------------------------------------+
|                                                                              |
|     The Edit Line Comment Create Function Options:                           |
|                                                                              |
|     1) The line comment left margin  (35 to 40) -  36   (Default = 36)       |
|     1) The line comment right margin (46 to 72) -  71   (Default = 71)       |
|     3) The comment head token      (max 3 char) -  /*   (Default = /*)       |
|     4) The comment tail token      (max 3 char) -  */   (Default = */)       |
|                                                                              |
|                                                                              |
|     Press PF01 for tutorial guide.                                           |
|     Press ENTER to display Balance Code Profile Group.                       |
|     Press END to return to program source file.                              |
|     Press CANCEL to cancel the option changes.                               |
|                                                                              |
+------------------------------------------------------------------------------+
```

On this option selection panel, you may change the value of some options for your own needs.

B.24 The 'CMTLINE' option selection panel on−line tutorial

If you press the PF1 key on the CMTLINE option selection panel, then the following tutorial screen will be displayed:

Figure 39. The introduction tutorial guide of the 'CL' edit command

```
+-------------------------------------------------------------------------------+
|                                                                               |
|                              Introduction                                     |
|                                                                               |
|     The 'CL' command can be used to create an empty remark comment line in    |
|     the edited file area. On this control panel, you may specify the left     |
|     and the right margins of the comment lines. You may also choose the       |
|     the comment head and tail tokens based on your own preference. The        |
|     default comment head and tail tokens are '/*' and '*/'. Thus, when a      |
|     'CL' edit command is executed, the following sample remark line command   |
|     will be created on a code line from column 36 to column 71:               |
|                                                                               |
|     If test_flag = 1 then             /* _                         */        |
|                                                                               |
|     The cursor will be automatically placed at the position as shown in       |
|     the above example, which allows you to enter the data very easily.        |
|     After the data is filled in and the Enter key is pressed, another         |
|     empty remark comment line will be automatically appended to this line.    |
|                                                                               |
|     To delete an existing remark comment line, you may enter a 'CL D'         |
|     command and type an 'A' on the prefix command line area.                  |
|                                                                               |
|                                                                               |
+-------------------------------------------------------------------------------+
```

If you press the Enter key or the PF8 key, then the following tutorial guide will be displayed:

```
--------------------------------------------------------------------------------

Note:              To display this 'CL' command options panel, you need to
                   enter a 'CL OPTION' or 'CL O' command on the command line of
                   the edited file. The short command form of the 'CL O' is
                   the 'CLO' command. You may enter a 'CL ?' command for the
                   detail information of the usages of this command.

Left Margin        The column number of the leftmost column of the remark line
                   comment. The valid range of the left margin is from column
                   35 to column 40. The default left margin is column 36.

Right Margin       The column number of the rightmost column of the remark line
                   comment. The valid range of the right margin is from column
                   46 to column 72. The default right margin is column 71.

Comment Head       The line comment head is the left comment delimiter of the
                   comment line. The valid comment head must be less than 3
                   characters. The default comment head token is '/*'. The
                   other commonly selected comment head tokens are '//', '/*!',
                   '/**', and '{', etc.

Comment Tail       The line comment tail is the right comment delimiter of the
                   comment line. The valid comment head must be less than 3
                   characters. The default comment head token is '*/'. The
                   other commonly selected comment tail tokens are '//', '**/',
                   and '}', etc. If you want to set the comment tail token to
                   blank, then you must enter a 'NUL' keyword in this field.

------------------ End of 'CL' edit command on-line tutorial -----------------
```

B.25 The examples of the 'CL' command

The following few diagrams can describe how the 'CMTLINE' edit command function works in the edit file. Assume a 'CL 8' edit command is entered at the command line or a 'CL' edit command is entered at the command line with an 'A' line command typed at line 8 in the sample C program which is as shown below:

Figure 40. The method of entering a 'cl 8' edit command in edit file

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                      Columns 00001 00072     |
|  Command ===> _cl 8_____________________________________ Scroll ===> CSR      |
|  ****** **************************** Top of Data ************************     |
|  000001  /* Hello.c - sample program */                                       |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                       |
|  000004  /*-----------------------------------------------------------------*/ |
|  000005  /* This is a sample C program named Hello.c                        */ |
|  000006  /*-----------------------------------------------------------------*/ |
|  000007  void main() {                                                        |
|  000008     printf("Goodbye old world\n");                                    |
|  000009     printf("Hello NEW world\n");                                      |
|  000010  }                                                                    |
+-------------------------------------------------------------------------------+
```

After pressing the Enter key, a new remark line comment will be created and attached to the end of the line 8, which is shown in the following diagram. Note that the cursor has been intelligently placed at a position in the remark comment line area of the code line 8 which has labeled with a '.CL' flag. The edited file command line will contain a 'CMTLINE' command code. This will bring the 'CL' edit command function into the 'Automatic Insertion Mode' within the CMTLINE mode.

Figure 41. A new empty remark comment line inserted at line 8

```
+-------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.C(HELLO)                      Columns 00001 00072     |
|  Command ===> CMTLINE_____________________________________ Scroll ===> CSR     |
|  ****** **************************** Top of Data ************************     |
|  000001  /* Hello.c - sample program */                                       |
|  000002  #include <stdio.h>                                                    |
|  000003                                                                       |
|  000004  /*-----------------------------------------------------------------*/ |
|  000005  /* This is a sample C program named Hello.c                        */ |
|  000006  /*-----------------------------------------------------------------*/ |
|  000007  void main() {                                                        |
|  .CL        printf("Goodbye old world\n"); /* _                          */ |
|  000009     printf("Hello NEW world\n");                                      |
|  000010  }                                                                    |
+-------------------------------------------------------------------------------+
```

If any input data is typed over the empty remark comment line area and the Enter key is pressed, then a new empty remark comment line will be inserted below line 8 and the cursor will be intelligently place at a position which allows you to continue to enter the text data in the next empty remark comment line in an Automatic Insertion Mode. The following diagram shows how it works:

Figure 42. Another empty remark comment line inserted below line 8

```
+-------------------------------------------------------------------------------+
```

```
|  EDIT       USERID.TEST.C(HELLO)                        Columns 00001 00072      |
|  Command ===> CMTLINE_____________________________________ Scroll ===> CSR       |
|  ****** **************************** Top of Data ************************        |
|  000001  /* Hello.c - sample program */                                          |
|  000002  #include <stdio.h>                                                       |
|  000003                                                                          |
|  000004  /*--------------------------------------------------------------------*/ |
|  000005  /* This is a sample C program                                         */ |
|  000006  /*--------------------------------------------------------------------*/ |
|  000007  void main() {                                                           |
|  000008     printf("Goodbye old world\n"); /* Say goodbye to old world  */     |
|  .CL                                      /* _                         */     |
|  000010     printf("Hello NEW world\n");                                        |
|  000011  }                                                                       |
+----------------------------------------------------------------------------------+
```

If no input data is typed over the empty remark comment line when the Enter key is pressed, then the empty remark comment line will be dropped and the CMTLINE mode will be terminated.

Instead of pressing the Enter key over an empty comment line to stop the CMTLINE mode, you may add an 'END' parameter to the 'CMTLINE' edit command, i.e. enter an 'CMTLINE END' command, to terminate the CMTLINE mode also.

Note: It is allowable to set the comment tail token to be a blank by using a 'NUL' keyword. Thus, for inserting the comment boxes and remark line comments in the C++ programs, it is highly recommended that you should set the CMTBOX and the CMTLINE options selection panels using the following example:

```
The Edit Box Comment Create Function Options:

1) The left margin of the box     (1 to 5) -  1    (Default =  2 )
2) The right margin of the box   (9 to 72) -  72   (Default =  71)
3) The comment frame code  (*, -, =, or /) -  -    (Default =  - )
4) The comment head token     (max 3 char) -  //   (Default =  /*)
5) The comment tail token     (max 3 char) -  //   (Default =  */)
```

```
The Edit Line Comment Create Function Options:

1) The line comment left margin  (35 to 40) -  35   (Default = 36)
1) The line comment right margin (46 to 72) -  72   (Default = 71)
3) The comment head token      (max 3 char) -  //   (Default = /*)
4) The comment tail token      (max 3 char) -  NUL  (Default = */)
```

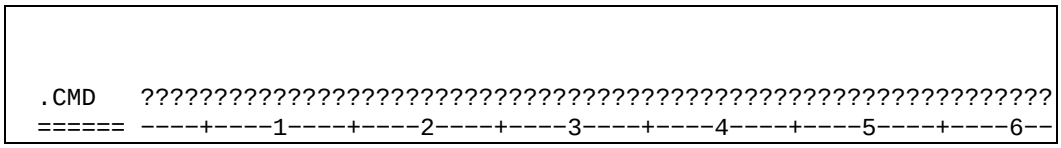
B.26 The descriptions of the 'COLUTIL' edit command

The 'COLUTIL' edit command can be used to perform various kinds of "ISPF Editor Column Utility" functions. The short command form of the 'COLUTIL' command is the 'COL' edit command. You may also type a pair of 'CXX' line command to perform various 'COLUTIL' functions.

The following document was extracted from the on−line tutorial guide of the 'COL' edit command when a 'COL ?' command is entered on an edited file command line:

The 'COLUTIL' is an Edit Macro. To apply this function, simply enter a 'COL' edit command on the edit command line and type a 'C' or a pair of 'CC' line command to specify the range of code lines that the column utility function will be performed. If both of the 'C' or 'CC' line commands are omitted, then the code lines ranging from the top line of the current displayed file screen to the end of the edited file will be selected.

After the 'COL' edit command is executed, a horizontal command bar and a column scale line, which is as shown below, will be displayed at the beginning of the selected code lines:



Where the ".CMD" is a label generated by this command. Two other labels ".BEGIN" and ".END" will also be generated and assigned to the first and last selected code lines of the 'COL' edit command function.

Now you may enter one of the following valid commands to choose the column zones and perform various types of the "ISPF Editor Column Utility" functions:

1. Insert − Use the "I" code on the CMD line to insert blank columns start at the selected column.
2. Delete − Use the "D" code on the CMD line to delete the selected columns.
3. Repeat − Use the "R" code on the CMD line to repeat the selected column zone.
4. Blank − Use the "B" code on the CMD line to blank out the selected column zone.
5. Sumup − Use the 'S' code on the CMD line to sum up the total numbers in the selected column zone.
6. Typeover − Use the "T" code followed by non−blnak data to type over on the selected column zone.
7. Copy − Use the "C" code to specify a selected column zone and use a "O" or "A" code to copy to the selected columns either override or not override the existing data on the target column zone.
8. Move − Use the "M" code to specify a selected column zone and use a "O" or "A" code to move to the selected columns either override or not override the existing data on the target column zone.
9. Swap − Use the "X" and "Y" code to specify two column zones and swap the data in two zones.

The command code "IIIII", for example, for inserting five columns can be simplified as a "I5" command. Similarly all other command codes can be simplified in the same manner. If the column amount is omitted, i.e. if you specify an "I" command on the horizontal command bar, then the default is to insert only one blank column.

Note that the column amount must immediately follow the "I" or other command code. Thus, "i5" is a valid code and the "i 5" is not a valid code. All command codes are not case sensitive. Multiple pairs of "I" command can be specified to insert several blank columns at the same time unless the new code is inserted. If the new code will be inserted, then only one "I" command will be executed each time. For example, in the

following horizontal command bar the second "i" command for inserting the "*/" code at column 45 will be ignored.

```
 .CMD   ?i/*????????????????????????????????????i*/??????????????
 ====== ----+----1----+----2----+----3----+----4----+----5----+----6--
```

For the 'Insert', 'Repeat', 'Copy At', and 'Move At' functions, the data longer than the logical record length of the code lines will be truncated.

If the "CCCCCC" or "MMMMMM" command code is specified for copying or moving the data on six columns after or override at the target location, for example, then either one of the "A" (At) and "O" (Override) target codes must be also specified.

Only one type of 'COL' edit command pattern can be specified each time. The different command codes cannot be mixed used on the horizontal command bar. Otherwise an error message will be prompted. When the error occurs, you may either blank out the extraneous code or simply type a code "Z" at the first column on the horizontal command bar to reset it.

To terminate the 'COL' edit command process, you may enter either a 'COLUTIL END' or enter a '/R' code at the first column on the horizontal command bar. If you don't type any command code on the horizontal command bar and press the Enter key, then the 'COL' edit command process will also be terminated.

If the horizontal command bar is left in your file by mistake after the COLUTIL mode is terminated due to program error, then simply delete this line from your file manually.

B.27 The examples of the 'COL' edit command

The following few diagrams can describe how the 'COLUTIL' edit command function works in the edit file. Assume a 'COL' edit command is entered at the command line and a pair of 'CC' line command is typed in at the prefix line command area of line 2 and line 9, which is as shown below:

Figure 43. The example of the 'COL' command with a pair of 'CC' line command

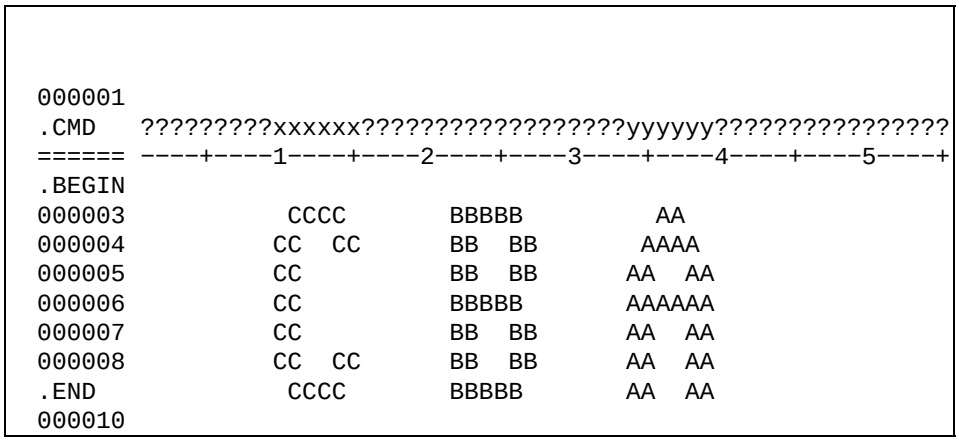
```
+------------------------------------------------------------------------------+
|  EDIT       USERID.TEST.FILE(SAMPLE)                 Columns 00001 00072     |
|  Command ===> _col_____________________________________ Scroll ===> CSR      |
|  ****** ***************************** Top of Data ***********************    |
|  000001                                                              */      |
|  CC 002                                                              */      |
|  000003           CCCC       BBBBB         AA                               |
|  000004          CC  CC      BB  BB       AAAA                              |
|  000005          CC          BB  BB      AA  AA                             |
|  000006          CC          BBBBB       AAAAAA                             |
|  000007          CC          BB  BB      AA  AA                             |
|  000008          CC  CC      BB  BB      AA  AA                             |
|  CC 009           CCCC       BBBBB       AA  AA                             |
|  000010                                                                      |
+------------------------------------------------------------------------------+
```

Note: The big letters 'C', 'B', and 'A' can be created in your edited file by using a 'BIGCHAR' or 'BIGC' Edit

Macro command.

If you want to swap the two big letters 'C' and 'A', then you may type the "X" and "Y" command codes on the horizontal command bar as follows:

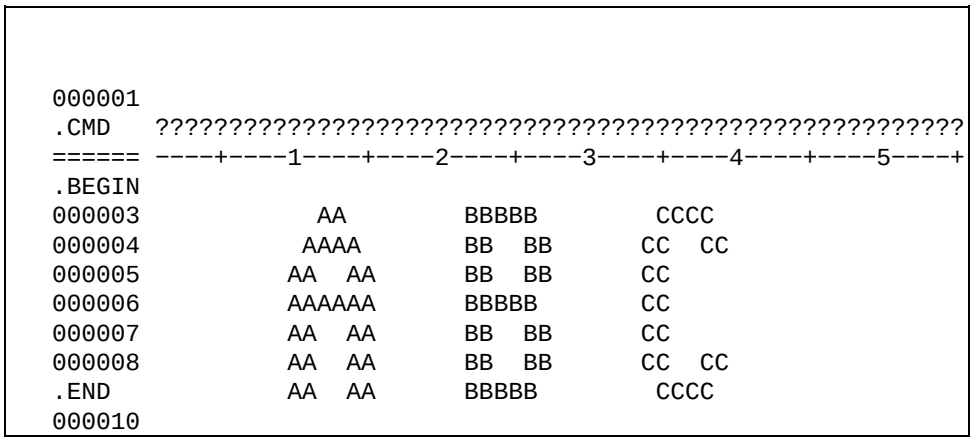
Figure 44. The 'X' and 'Y' command codes on the horizontal command bar



Note: By observing the above diagram, you can see that only one 'X' column zone and one 'Y' column zone can be specified on the horizontal command bar each time. The 'X' and 'Y' command code must be specified in a pair.

After pressing the Enter key, the following is the result:

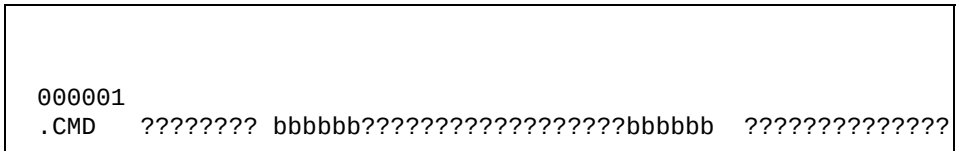
Figure 45. The two columns are swapped from the 'X' and 'Y' command codes



Note that the two column zones need not be in the same size. However it will affect the data location in the middle of the two column zones after the swapping if the sizes of the two column zones are different.

Similarly, if you want to blank out the two big letters 'A' and 'C', then you may type the "B" command codes on the horizontal command bar as follows:

Figure 46. The 'B' command code can be used to blank out data



```
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN
000003            AA         BBBBB         CCCC
000004           AAAA        BB   BB      CC  CC
000005          AA  AA       BB   BB      CC
000006          AAAAAA       BBBBB        CC
000007          AA  AA       BB   BB      CC
000008          AA  AA       BB   BB      CC  CC
.END            AA  AA       BBBBB         CCCC
000010
```

Note: By observing the above diagram, you can see that the 'B' command code can be specified on several column zones on the horizontal command bar each time. But no other command code can be mixed with the 'B' command code. The blank space on the horizontal command bar will be treated the same as the '?' code and it will be ignored.

After pressing the Enter key, the following is the result:

Figure 47. The two columns are blanked out from the 'B' command code

```
000001
.CMD   ??????????????????????????????????????????????????????
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN
000003                      BBBBB
000004                      BB   BB
000005                      BB   BB
000006                      BBBBB
000007                      BB   BB
000008                      BB   BB
.END                        BBBBB
000010
```

Now you may enter a 'T' command code with a input code "/*!", which is shown as follows:

Figure 48. The 'T' command code with a input code '/*!' at column 1

```
000001
.CMD   t/*!?????????????????????????????????????????????????
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN
000003                      BBBBB
000004                      BB   BB
000005                      BB   BB
000006                      BBBBB
000007                      BB   BB
000008                      BB   BB
.END                        BBBBB
000010
```

Then after pressing the Enter key, the '/*!' code will be typed over in the edited file, which is show as follows:

Figure 49. The result of typing the input code '/*!' over at column 1

```
000001
.CMD   ????????????????????????????????????????????????????
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN /*!
000003 /*!                  BBBBB
000004 /*!                  BB  BB
000005 /*!                  BB  BB
000006 /*!                  BBBBB
000007 /*!                  BB  BB
000008 /*!                  BB  BB
.END   /*!                  BBBBB
000010
```

Note that the input code '/*!' is over typed at the column zone start from column 1 but not from column 2.

You may enter the 'CC' command code on column 1 and column 2 with an 'A' target code to copy the data '/*' from column 1 to column 50, which is shown as follows:

Figure 50. The 'C' command code with an 'A' code is used to copy the data

```
000001
.CMD   cc  ???????????????????????????????????????????a?????
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN /*!
000003 /*!                  BBBBB
000004 /*!                  BB  BB
000005 /*!                  BB  BB
000006 /*!                  BBBBB
000007 /*!                  BB  BB
000008 /*!                  BB  BB
.END   /*!                  BBBBB
000010
```

After pressing the Enter key, the following is the result:

Figure 51. The code '/*' at column 1 is copied over at column 50

```
000001
.CMD   ????????????????????????????????????????????????????
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN /*!                                                  /*
000003 /*!                  BBBBB                           /*
000004 /*!                  BB  BB                          /*
000005 /*!                  BB  BB                          /*
000006 /*!                  BBBBB                           /*
000007 /*!                  BB  BB                          /*
000008 /*!                  BB  BB                          /*
.END   /*!                  BBBBB                           /*
000010
```

Assume the 'M A' command code is entered at column 50, which is shown as follows:

Figure 52. The 'M' command code is used to move the data on column 50

```
000001
.CMD   ?????????????????????????????????????????????m a???
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN /*!                                                  /*
000003 /*!                   BBBBB                          /*
000004 /*!                   BB  BB                         /*
000005 /*!                   BB  BB                         /*
000006 /*!                   BBBBB                          /*
000007 /*!                   BB  BB                         /*
000008 /*!                   BB  BB                         /*
.END   /*!                   BBBBB                          /*
000010
```

Then after pressing the Enter key the '/' code at column 50 will be moved and column 50 it has new code '*/' instead, which is as shown in the following diagram:

Figure 53. The new code '*/' is now located at column 51

```
000001
.CMD   ?????????????????????????????????????????????????
====== ----+----1----+----2----+----3----+----4----+----5----+
.BEGIN /*!                                                  */
000003 /*!                   BBBBB                          */
000004 /*!                   BB  BB                         */
000005 /*!                   BB  BB                         */
000006 /*!                   BBBBB                          */
000007 /*!                   BB  BB                         */
000008 /*!                   BB  BB                         */
.END   /*!                   BBBBB                          */
000010
```

Note: The 'C' (Copy) and 'M' (Move) command code must be paired up with either an 'A' (At) or an 'O' (Override) code on the horizontal command bar each time. The 'A' or 'O' code can be specified on the column that is on the left to the 'C' or 'M' command code when the column data will need to be copied or moved to the left column zone.

Finally, after all these functions are completed, you may press the Enter key again without specifying any command code to exit the 'COLUTIL' edit command process. Meanwhile, the horizontal command bar, the tool generated scale line, and the '.BEGIN' and '.END' labels will all be dropped. If you delete any of these lines by mistake before the 'COLUTIL' edit command process is completed, then the unexpected result will occur.

As you have seen from the above examples, it is very restrictive that the rules of this Edit Macro must be followed and the different types of command codes cannot be mixed. Otherwise, unexpected result might occur.

Note: The following is the summary of several important points of this 'COLUTIL' edit command functions:

1. The valid command codes that are allowed to be entered on the horizontal command bar are: 'I', 'D', 'R', 'B', 'S', 'T', 'C' 'M', 'X', 'Y'.

2. Only the 'I', 'D', 'R', and 'B' command codes can be specified on several column zones, the rest command codes must be specified on a single column zone.

3. The 'C' and 'M' command codes must be paired up with the 'A' or 'O' target code.

4. The 'T' command code must be followed by one and only one input character string.

5. The 'I' command code followed by an input character string is very similar to the 'T' command. The difference of these two types of command codes is that the character string in 'T' command will be typed over at the selected column of the edited file and the character string in the 'I' command will be inserted in the selected column of the edited file.

6. The different types of command codes cannot be mixed used on the same horizontal command bar each time. Otherwise, an error message of "Extraneous command code" will be issued.

7. When the error occurs, the 'Z' command code can be used to reset the horizontal command bar back to its original state.

8. Instead of entering a 'COLUTIL END' command, you may type a '/R' command code on the horizontal command bar to terminate the 'COLUTIL' edit command process also.

9. Both of the 'Z' and '/R' command codes must be typed at the first column on the horizontal command bar.

10. Please don't delete the tool generated code lines while processing the Edit Macro commands. Otherwise it will generate the unexpected result. In case you do delete these code lines by mistake, then you may enter a 'CANCEL' command to abort the COLUTIL process.

B.28 The descriptions of the 'SORTUTL' edit command

The 'SORTUTL' edit command can be used to perform the ISPF SORT edit command function. The short command form of the 'SORTUTL' edit command is the 'SRT' edit command.

The following document was extracted from the on−line tutorial guide of the 'SRT' edit command when a 'SRT ?' command is entered on an edited file command line:

The 'SORTUTL' is an Edit Macro. To apply this function, simply enter a 'SRT' command on the edit command line and type a pair of 'CC' line command to specify the range of code lines that the SORT command function will be performed. If the "CC" line command is omitted, then the code lines ranging from the top line of the current displayed file screen to the end of the edited file will be selected.

You may specify either a "X" or a "NX" option code with the "SRT" edit command to sort only the exclusive or non−exclusive lines. This option code can be entered either with the 'SRT' edit command or with the 'SORTUTIL' edit command. To specify this code is optional.

After the 'SRT' edit command is executed, a horizontal command bar and a column scale line, which is as shown below, will be displayed at the beginning of the selected code lines:

```
  .CMD   ????????????????????????????????????????????????????????
  ====== ----+----1----+----2----+----3----+----4----+----5----+
```

Where the ".CMD" is a label generated by this command. Two other labels ".BEGIN" and ".END" will also be generated and assigned to the first and last selected code lines of the 'SRT' edit command function.

Now you may enter either the code "A" or "D" on the horizontal command bar to specify the column zones for either sorting in ascending or descending order. If the column zone range is too large, you may specify the size of the column zone immediately follow the "A" or "D" code.

For example, if the following command pattern is specified:

```
??????????aaaaaaaaaaa????d?????????aaaaaa????d14???????????
<----+----1----+----2----+----3----+----4----+----5----+-->-
```

Then this macro will translate the above command pattern into the following SORT command:

```
"SORT .BEGIN .END A 10 20 D 25 25 A 35 40 D 45 58"
```

If you want to use the whole length of the record to be selected as the column zone and sort the file in ascending order, then you may specify a "A99" code as follows:

```
a99????????????????????????????????????????????????????????
<----+----1----+----2----+----3----+----4----+----5----+-->-
```

Then this macro will translate this command pattern into the following SORT command:

```
"SORT .BEGIN .END A 1 58"
```

Note that the column amount must immediately follow the "A" or "D" code. Thus, the "a99" is a valid code and the "a 99" is not a valid code. All command codes are not case sensitive.

Note also that the left and right bounds of this edited file are between column 1 and column 58. Any command code that you specify on the horizontal command bar must be in this bounds.

The macro can perform the function to remove the duplicated code lines after the sorting by specifying the code "N". For example, if the following command pattern is specified:

```
??????????nnnnnnnnnnn????n?????????nnnnnn????nnnn???????????
<----+----1----+----2----+----3----+----4----+----5----+-->-
```

Then this macro will translate this command pattern into the following NODUP command:

```
"NODUP .BEGIN .END 10 20 25 25 35 40 45 48"
```

Note: The NODUP command will be described in details at the next section.

Only one type of 'SRT' edit command pattern can be specified each time. The "A" and "D" command codes cannot be mixed used with the "N" command code on the horizontal command bar. Otherwise an error message will be prompted. When the error occurs, you may either blank out the extraneous code or simply type a code "Z" at the first column to reset the horizontal command bar.

To terminate the 'SRT' edit command process, you may enter either a 'SORTUTL END' or a enter a '/R' code at the first column on the horizontal command bar. If you don't type any command code on the horizontal command bar and press the Enter key, then the 'SRT' edit command process will also be terminated.

If the horizontal command bar is left in your file by mistake due to program error, then simply delete this line from your file manually.

B.29 The descriptions of the 'NODUP' edit command

The 'NODUP' edit command can be used to perform the duplicated code lines removal function in edit file.

The following document was extracted from the on−line tutorial guide of the 'NODUP' edit command when a 'NODUP ?' command is entered on an edited file command line:

The 'NODUP' is an Edit Macro. To apply this function, simply enter a 'NODUP' edit command on the edit command line and type a pair of 'CC' line command to specify the range of code lines that the NODUP edit command function will be performed. If the "CC" line command is omitted, then the code lines ranging from the top line of the current displayed file screen to the end of the edited file will be selected.

After the 'NODUP' edit command is executed, a horizontal command bar and a column scale line, which is as shown below, will be displayed at the beginning of the selected code lines:

```
.CMD   ????????????????????????????????????????????????????
====== ----+----1----+----2----+----3----+----4----+----5----+
```

Where the ".CMD" is a label generated by this command. Two other labels ".BEGIN" and ".END" will also be generated and assigned to the first and last selected code lines of the 'NODUP' edit command function.

Now you may enter the code "N" on the horizontal command bar to specify the column zones for the duplicated lines removal functions. If the column zone range is too large, you may specify the size of the column zone immediately follow the "N" code.

For example, if the following command pattern is specified:

```
??????????nnnnnnnnnnn????n??????????nnnnnn????n14???????????
<----+----1----+----2----+----3----+----4----+----5----+-->-
```

Then this macro will translate this command pattern into the following NODUP command:

```
"NODUP .BEGIN .END 10 20 25 25 35 40 45 58"
```

If you want to use the whole length of the record to be selected as the column zone, then you specify a "N99" code as follows:

```
n99?????????????????????????????????????????????????????????
<----+----1----+----2----+----3----+----4----+----5----+-->-
```

Then this macro will translate this command pattern into the following NODUP command:

```
"NODUP .BEGIN .END 1 58"
```

Note that the column amount must immediately follow the "N" code. Thus, the "n99" is a valid code and the "n 99" is not a valid code. All command codes are not case sensitive.

Note also that the left and right bounds of this edited file are between column 1 and column 58. Any command code that you specify on the horizontal command bar must be in this bounds.

Only one type of 'NODUP' edit command pattern can be specified each time. The "N" command code cannot be mixed used with any other characters on the horizontal command bar. Otherwise an error message will be prompted. When the error occurs, you may either blank out the extraneous code or simply type a code "Z" at the first column to reset the horizontal command bar.

To terminate the 'NODUP' edit command process, you may enter either a 'NODUP END' or a enter a '/R' code at the first column on the horizontal command bar. If you don't type any command code on the horizontal command bar and press the Enter key, then the 'NODUP' edit command process will also be terminated.

If the horizontal command bar is left in your file by mistake due to program error, then simply delete this line from your file manually.

Appendix C. How to use the 'UPDPDS' command to update the PDS libraries

If you make a copy of a tools PDS library to your TSO account and modified some members, and you wish update this tool library based on your modification, but you don't want to copy every member from the PDS on your TSO account to that library, then you may use this 'UPDPDS' command to perform the task.

The 'UPDPDS' command can be used to compare two PDS libraries and create a Incremental Delta PDS file, in which it contains the members that have been revised, for the PDS library update function. The following is the descriptions of the method how to use this command on the PANEL3 panel in the diagram format.

C.1 The description of the 'UPDPDS' command

Assume that you have released this PROJWRK tool package on the 'TOOLKIT' library system, and your PANEL3 panel of the 'PROJECT' project has defined as follows:

Figure 54. The sample PANEL3 panel with the 'UPDPDS' command

```
+-------------------------------------------------------------------------------+
|  PANEL3       The Data Set or Command Selection Panel          Row 1 to 15 of 1|
|  ---------------------------------------------------------------------------- |
|  Project Code ===> 4       Project Name ===> PROJECT            Time => 12:49:1|
|  Function     ===> The MVS Project Work Manager                               |
|  Command ===> _____________________________________________  Scroll ===> CSR|
|                                                                               |
|   Select    Code    PDS, Sequential, GDG, VSAM, or TSO/ISPF commands     Volume|
|  -------- -------- ------------------------------------------------- ------|
|  _________     1     TOOLKIT.@PROJWRK.PACKAGE                                 |
|  _________     2     TOOLKIT.@PROJWRK.ANNOUNCE                                |
|  _________     3     TOOLKIT.@PROJWRK.CEXEC                                   |
|  _________     4     TOOLKIT.@PROJWRK.LOAD                                    |
|  _________     5     TOOLKIT.@PROJWRK.PANELS                                  |
|  _________     6     TOOLKIT.@PROJWRK.SKELS                                   |
|  _________     7     TOOLKIT.@PROJWRK.TABLE                                   |
|  _________     8     TOOLKIT.@PROJWRK.DOCUMENT                                |
|  _________     9     USERID.@PROJWRK.LIST                                     |
|  _________    10     USERID.@PROJWRK.XREF                                     |
|  _updpds_____    11     USERID.@PROJWRK.CEXEC                                |
|  _________    12     USERID.@PROJWRK.PANELS                                   |
|  ****************************** Bottom of data ******************************|
|                                                                               |
+-------------------------------------------------------------------------------+
```

On the above panel, you have defined two data set name entries of the CEXEC and PANELS libraries of this tool package. Suppose you have just recently received the revised CEXEC and PANELS program code from the MVSTOOLS library system and stored these two new PDS on your own TSO session, and you want to update these two libraries on the 'TOOLKIT' library system, then you may use the 'UPDPDS' command to create a Delta PDS in which it contains the member files that contain the revised code.

After the Enter key is pressed, the following window panel will be displayed:

Figure 55. The sample window panel for generating a Delta PDS

```
+-------------------------------------------------------------------------------+
```

```
|                                                                              |
|     ------ Compare two PDS files and create a Delta PDS file -------         |
|                                                                              |
|     Source PDS  . . . :  'USERID.@PROJWRK.CEXEC'                             |
|     Volume  . . . . . :                                                      |
|                                                                              |
|     Enter the Target PDS name for Comparison:                                |
|                                                                              |
|     Target PDS  . . . :  'TOOLKIT.@PROJWRK.CEXEC'                            |
|     Volume  . . . . . :                                                      |
|                                                                              |
|     Delta PDS . . . . :  'USERID.@@DELTA.@PROJWRK.CEXEC'                     |
|                                                                              |
|     Press PF01 for tutorial guide.                                           |
|     Press PF11 to display Data Set List for selection.                       |
|     Press ENTER to confirm comparison function.                              |
|     Press END or CANCEL to cancel comparison function.                       |
|                                                                              |
+------------------------------------------------------------------------------+
```

On this panel, originally the data set names of both Source and Target PDS are identical. You will need to modify the Target PDS data set name before pressing the Enter key. The Delta PDS name is generated by this tool and it cannot be changed.

Note: Instead of modify the Target PDS name manually on the above panel, on the sample PANEL3 panel you may type a 'S' (Select) command code next to the third data set name, i.e. the 'TOOLKIT.@PROJWRK.CEXEC' file, at the same time when you type the 'UPDPDS' command next to the eleventh data set name, i.e. the 'USERID.@PROJWRK.CEXEC' file.

After the Enter key is pressed on this panel, the Delta PDS will be generated on your TSO session.

C.2 The method of updating the PDS library

Now you enter a 'FFF' command on the PANEL3 panel to display a 'DSLISTF' panel, which is shown as follows:

Figure 56. The sample ISPF option 3.4 Front−end Interface 'DSLISTF' panel

```
+------------------------------------------------------------------------------+
|  DSLISTF      The Data Set List Front-end Interface panel       Row 1 to 9 of 9|
|  ----------------------------------------------------------------------------|
|  Command ===> _e_____________________________________________  Scroll ===> CSR|
|                                                                              |
|  Select  Code  Dsname Level codes                 Descriptions          Volume|
|  ------ ----- ---------------------------------- ---------------------- ------|
|  _______    1 - USERID.*                         My data sets                 |
|  _______    2   SYS1                             SYS1 data sets               |
|  _______    3   TOOLKIT                          TOOLKIT files                |
|  _______    4   TOOLKIT.@PROJWRK                 PROJWRK libraries            |
|  _______    5   MXG.*                            The MXG stuff         DSSPK0 |
|  _______    6   SAS.*                            SAS tools                    |
|  ****************************** Bottom of data ******************************|
+------------------------------------------------------------------------------+
```

Assume that a Last−Used marker "−" has already been flagged on the 'USERID.*' entry, then after the Enter key is pressed all of your TSO cataloged data set names will be displayed on the ISPF option 3.4 Data Set List panel, which including the following two data set names:

Figure 57. The sample Delta PDS files

```
USERID.@@DELTA.@PROJWRK.CEXEC
USERID.@@DELTA.@PROJWRK.PANELS
```

On the Data Set List panel, next to these two Delta PDS you may type a 'CPY' command to invoke the ISPF Copy/Move Utility program and display the ISPF option 3.3 panel to update the 'TOOLKIT.@PROJWRK.CEXEC' and 'TOOLKIT.@PROJWRK.PANELS' libraries.

After this PDS library update process is completed, you may delete these two Delta PDS.

After this PDS library update process is completed, you should delete the two CEXEC and PANELS library files from your TSO session also. Otherwise, it will issue an error message when you try to enter the PROJWRK tool session again.

The advantage of using the 'UPDPDS' command to generate the Delta PDS for updating the Target PDS instead of using the revised PDS itself to update the Target PDS is that it can save the space of the Target PDS so that you don't need to compress it often whenever it is updated.

Appendix D. How to use the 'CPYFROM' and 'HEXTFROM' commands

There is a 'CPYFROM' command available for you to copy a very large file partially to a new created sequential file in either foreground mode or in a batch job. The short form of this command is the 'CPYF' command.

Similarly, if you want to copy a very large text file partially to a new created sequential file in hexadecimal type format, then there is a 'HEXTFROM' command available. The short form of this command is the 'HEXTF' command.

Both of these two commands can be invoked either next to a sequential file on the PANEL3 panel or next to a PDS member file on the PANEL4 panel.

D.1 The description of the 'CPYFROM' command

Assume the 'CPYF' command is entered next to the 'TESTDATA' member file of a PDS, say the 'USERID.TEST.DUMP' file, on the PANEL4 panel, then the following sample window panel will be displayed:

Figure 58. The sample window panel for the CPYF command

```
+-------------------------------------------------------------------------------+
|                                                                               |
|       Source Data Set . . :  'USERID.TEST.DUMP(TESTDATA)'                     |
|                                                                               |
|       Enter the new created large target sequential data set name:            |
|                                                                               |
|       Target Data Set . . :  'USERID.@CPYFROM.TESTDATA.PDSMBR48'              |
|       Copy from the line  :  2001      (optional)                             |
|       For how many lines  :  5000      (optional)                             |
|                                                                               |
|                                                                               |
|       Press PF01 for tutorial guide.                                          |
|       Press PF04 to perform copy file function in foreground.                 |
|       Press ENTER to submit a batch job to copy the large file.               |
|       Press END or CANCEL to cancel the copy function.                        |
|                                                                               |
+-------------------------------------------------------------------------------+
```

On this panel, you may enter the data in the 'FROM' and 'FOR' option fields to specify the code lines range for the 'CPYFROM' command to copy to the target sequential file. In the above diagram, the FROM 2001 FOR 5000 sample code data has already entered, which means that the code lines start from line 2001 to line 7000 will be copied to a new sequential file. If the data in these two fields are omitted, then the default is to copy the entire source file to the target file.

A temporary target file named 'USERID.@CPYFROM.TESTDATA.PDSMBR48', for example, has already been generated by this tool and displayed on the panel. You may alter this field to a new file name as you wish. However, if it is modified then it must be a data set name of a non−existing file. To execute this command, you may press the Enter key on this window panel to submit a batch job. If you prefer to execute this command in foreground, then you may press the PF4 key instead of pressing the Enter key. After the 'CPYF' generated target file is created, you may enter a 'CPYF /' command on any process panel command line to display the target file in a data set list panel. Then on that panel, you may edit, browse, view, or rename the target file.

Note: Usually it is very difficult to compare two very large files by using the ISPF option 3.13. Thus, you may compare two very large files partially. To do that, first of all you may use the 'CPYF' command to create two new sequential files in batch jobs. After the two new files are created, then you may enter a 'FFF' command on any process panel to display the two new files on the ISPF option 3.4 Data Set List panel. Then from that panel you may type a 'DIFF' command next any of the two files to display a 'SuperCE Utility Panel' to compare the two new created files.

D.2 The description of the 'HEXTFROM' command

The method of creating a hexadecimal type of a very large file partially is very similar to the 'CPYFROM' command method. Assume the 'HEXTF' command is entered next to the 'TESTDATA' member file of a PDS, say the 'USERID.MAPMVS.LIST' file, on the PANEL4 panel, then the following sample window panel will be displayed:

Figure 59. The sample window panel for the HEXTF command

```
+-------------------------------------------------------------------------------+
|                                                                               |
|       Source Data Set . . :  'USERID.MAPMVS.LIST(TESTDATA)'                   |
|                                                                               |
|       Specify the range of code lines to be generated in HEXT code:           |
|                                                                               |
|       Start from the line :  101       (optional)                           |
|       For how many lines  :  2000      (optional)                           |
|                                                                               |
|                                                                               |
|       Press PF01 for tutorial guide.                                          |
|       Press PF04 to perform the HEXT code generation in foreground.           |
|       Press ENTER to submit a batch job to generate the HEXT code.            |
|       Press END or CANCEL to cancel the HEXT generation function.             |
|                                                                               |
+-------------------------------------------------------------------------------+
```

On this panel, you may enter the data in the 'FROM' and 'FOR' option fields to specify the code lines range for the 'HEXTFROM' command to generate a hexadecimal type of target sequential file. In the above diagram, the FROM 101 FOR 2000 sample code data has already entered, which means that the code lines start from line 101 to line 2100 will be copied to a new sequential file in hexadecimal type of code format. If the data in these two fields are omitted, then the default is to generate the hexadecimal code of the entire source file in the target file.

The generated target file of this command is always named 'USERID.@HEXTYPE.DATA'. To execute this command, you may press the Enter key on this window panel to submit a batch job. If you prefer to execute this command in foreground, then you may press the PF4 key instead of pressing the Enter key. After the 'HEXTF' generated target file is created, you may enter a 'HEXTF /' command on any process panel command line to display the target file in a data set list panel. Then on that panel, you may edit, browse, view, or rename the target file.