Fundamentals of TSO

Table of Contents

Fundamentals of TSO

Table of Contents

Fundamentals of TSO

1. INTRODUCTION

1.1 Abstract

TSO (Time Sharing Option) provides an interactive environment on the Academic IBM MVS system for program execution (interactive and batch), editing, printing, and managing data. Much of its functionality is provided by menu-driven full-screen utilities. This document describes how to use TSO and its major components.

1.2 General Information

Information regarding IOF, JCL, printing, disk quotas, and the url for online help has been updated.

2. OVERVIEW

TSO is available on the Academic IBM MVS system. Access is made via a tn3270 telnet session. ("tn3270" is a generic reference to a 3270-type terminal emulator. The standard telnet program usually does not emulate this terminal type. A special telnet program which emulates a terminal in the 327x family (and which often has "3270" in its name) is needed.)

Note: The same methods, software, userids and passwords formerly used to access full-screen WYLBUR are used to access TSO.

TSO is fairly simple to use. A system of menus is provided from which you select the tasks to be accomplished. These tasks include using the editor, submitting and monitoring batch jobs, interactive program execution, printing, and managing files. The names of the major components which provide these services are ISPF (the menu system), SPF/PDF (the editor), IOF (batch job monitor), and RACF (account attributes and security).

2.1 Major Components

ISPF (Interactive System Productivity Facility) is a menu-driven shell which simplifies TSO usage and provides numerous utility functions. The utility functions include creating, editing, deleting, printing, and renaming files, and looking at files, file directories, and catalog entries. ISPF is also used as an interface to access other software

on the system and often is required for interactive execution. ISPF is typically accessed by entering the command **ISPF**.

SPF/PDF (Program Development Facility) is a full-screen interactive text editor. It can also be used to submit batch jobs. The editor is typically accessed from ISPF utilities/dslist menu, but can also be accessed from the ISPF primary menu.

IOF (Interactive Output Facility) is a full-screen interactive utility for monitoring batch jobs, viewing batch job output, and purging or printing batch job output. IOF is typically accessed from the ISPF primary menu.

RACF (Resource Access Control Facility) controls disk and tape file security, keeps track of your disk space quota, default printer destination, logon password, and several other account related attributes. RACF is typically accessed from the ISPF primary menu.

2.2 Typical Terminal Session

The typical session begins with a tn3270 telnet connection using the address "acmvs.inre.asu.edu". As soon as the communications link with the IBM system is established, a VTAM System Selection screen displays a list of systems and sub-systems available for logon. Select Academic TSO to start the logon process.

Thereafter, you are prompted for a userid and then for a logon password and other optional information. One of optional items you can designate is a TSO command to be automatically executed as part of the logon process. Some people enter the **ISPF** command to start up ISPF immediately upon logon; others use the **EXEC** command to execute a program or exec (typically in REXX) to customize their logon process.

If you logon without entering an optional command, you are placed in TSO native mode, easily identifiable by a **READY** prompt. While in TSO mode, you can enter any TSO command. Most people enter the command **ISPF** to activate the menu-driven full-screen facilities. While it is not necessary to activate ISPF before doing useful work, some software require ISPF to be active for proper functioning. In particular, the full-screen text editor PDF is not available unless ISPF is active.

After ISPF is active, the primary ISPF menu is displayed. You can select the EDIT option on the primary menu (or in the UTILITIES/DSLIST) option to edit files and to submit batch jobs. Use the UTILITIES option to create, edit, delete, rename, and print files. Use the IOF option to monitor batch jobs. Use the EXIT option to exit ISPF and return to native TSO mode. (You must exit ISPF before the LOGOFF command will function.)

To select an ISPF option, type the option's corresponding identifier (a letter, number, or word) on the ISPF command line and press **ENTER**. Some options cause other menus

to appear which operate in the same fashion as the primary menu. Some of the menus may contain fields in which you need to type values, such as a data set name, before proceeding. As a general rule, after you've selected the appropriate options and have completed the desired tasks, you can return to the primary ISPF menu by pressing the **PF3** key one or more times.

To exit ISPF and return to native TSO mode, select the **X** option on the primary ISPF screen or enter the command **=X** from other ISPF screens. After you've returned to native TSO, you can enter other TSO commands, re-enter ISPF, or logoff.

3. ACCESSING THE SYSTEM

3.1 System Selection

```
                Welcome to ASU Computing Services

         * This system is only for use authorized by ASU *


                                                        STATUS

   1.  HELP
   2.  Scheduled Service Interruptions
   4.  VM/CMS - Information Center (Node ASUACAD)     ACTIVE
   8.  Academic TSO (Node ACADMVS)                    ACTIVE
  12.  Next Menu


Request ===>                Enter 1-12 or Press PF1-PF12 key
```

This is a **sample** System Selection screen. The actual screen to which you have access may differ substantially from that shown above; however, the format will be similar.

To select the Academic TSO system:

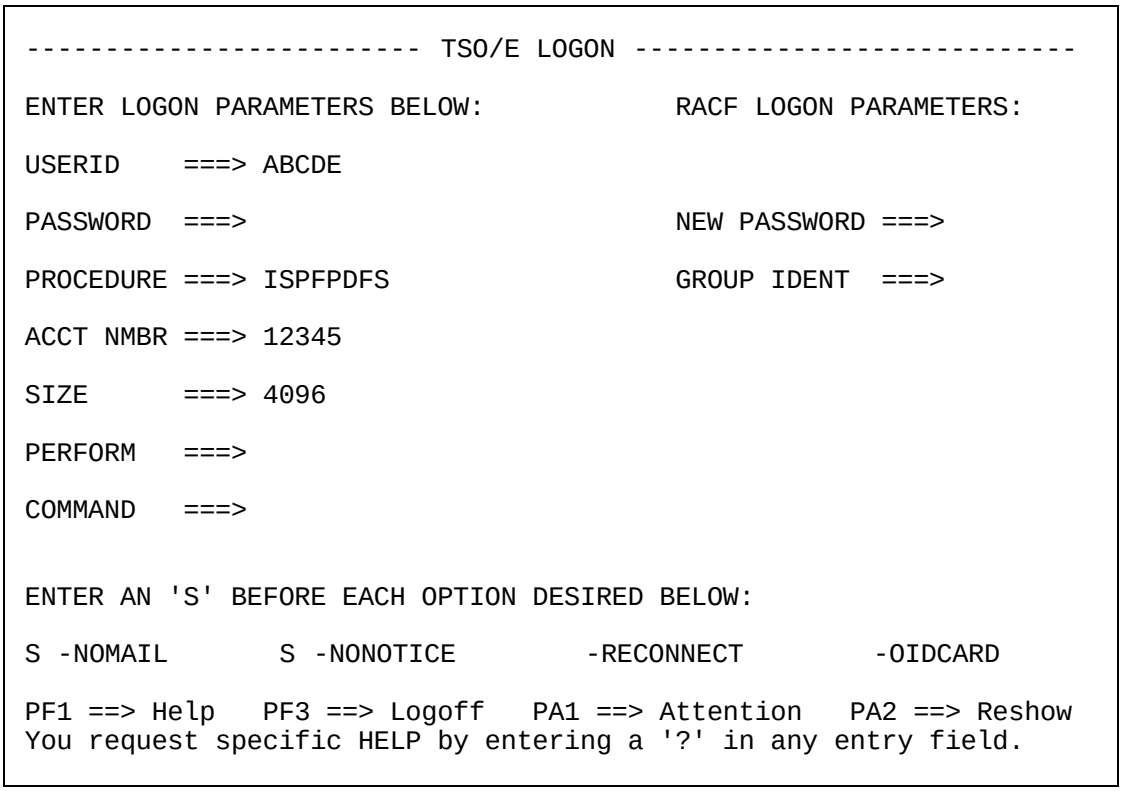
?? Determine which item on the screen corresponds to Academic TSO (in this sample, 8).
?? Type the corresponding number (8 in this sample) in the input field after the '===>' near the bottom of the screen, then press **ENTER**.

After Academic TSO has been successfully selected, the following prompt will appear:

```
IKJ56700A ENTER USERID -
```

Type your userid in response to this prompt and then press **ENTER**. The system will subsequently prompt for a logon password (and other optional information) on the TSO **Logon** screen.

3.2 TSO Logon



This is a **sample** TSO Logon screen. The actual screen to which you have access may differ and may have different values for the various fields; however, the format will be similar.

The sample illustrates possible logon values for userid ABCDE. The values displayed are those which were set when the account was established or which the owner of the userid set during a previous logon. (HELP information for any input field may be obtained by typing **?** in the input field and pressing **ENTER**. You may also request help by pressing the **PF1** key.)

You may type one or more values on this screen. Use the **TAB** key (not the **ENTER** key) to move from one field to another. Use the **ENTER** key after you've filled in the relevant fields.

As a general rule, you only need to type your logon password in the PASSWORD field and press **ENTER**. The first time you logon, type **ISPF** in the COMMAND field before pressing **ENTER**. (If you are using a self-contained application such as SAS, you may want to enter a command to start SAS instead.) If it's also the first time you've used the MVS system, type a new logon password in the NEW PASSWORD field. (Please remember your new password; you need it to logon next time.)

The **USERID** field contains the userid you entered in response to a logon prompt on a previous screen. If the userid is incorrect, press the **PF3** key to restart the logon process.

If the displayed userid is correct, type the logon password in the **PASSWORD** field. The password will be invisible.

The **NEW PASSWORD** field is used to change your logon password. Do not type anything in this field for any other purpose. The new password will be invisible. New passwords have a minimum size of 6 characters, a maximum of 8; contain uppercase alphabetic characters (**A**...**Z**), digits (**0**...**9**), and national characters (**$#@**); and cannot be identical to one of the recently used logon passwords. You are required to change your password the first time you use the MVS system and periodically thereafter. The system will prompt you to enter a new password when necessary.

> **CAUTION**: TYPE YOUR NEW PASSWORD CAREFULLY and REMEMBER IT.
> If you incorrectly type the new password or forget it, you will not be able to do future logons. In the event you are unable to determine the password for the userid, contact the Computer Accounts Office.

The standard value for **PROCEDURE** on the Academic TSO system is **ISPFPDFS**. This cannot be changed unless the userid is authorized for other procedures. The procedure allocates the standard files necessary for interactive use of the various software packages. You can customize the logon procedure, to allocate other files or to perform other functions, by typing an EXEC command in the COMMAND field (described below).

The standard value for **ACCT NMBR** is the userid's **account number**. This cannot be changed unless the userid is authorized for other account numbers.

The standard value for **SIZE** is **4096**, which requests 4 MB of virtual memory for your TSO terminal session. Each multiple of 1024 represents 1 MB. If the value is smaller or larger than your needs, type a new value. This field cannot be changed to a value greater than the userid's maximum for an interactive session. All userids are set up with a maximum of 32 MB for interactive processing. Batch jobs can be used to run programs which have larger memory requirements.

The **COMMAND** field is used to execute a single TSO command as part of the logon process. Leave the field **blank** to begin your terminal session in native TSO mode. To begin in full-screen ISPF mode, type **ISPF** in the field To customize the logon process, type an **EXEC** command in the COMMAND field to execute an exec you've written in either REXX (recommended) or CLIST; the exec may contain other TSO commands and may execute other execs and programs. To immediately invoke an application (for example SAS) without going into native TSO or ISPF mode, type the command which invokes the application.

The **PERFORM**, **GROUP IDENT**, and **OIDCARD** fields should be **blank**. These fields cannot be changed unless the userid is authorized for other values.

The **NOMAIL** and **NONOTICE** fields should be **S** to disallow message reception during logon or should be **blank** to allow receipt of messages during logon. Possible

messages that may be received include notification of file transfer and notification of batch job completion. It is recommended that you periodically put blanks in these two fields.

The **RECONNECT** field should be **blank** unless you are trying to reconnect to a disconnected session, in which case type **S** in the field. A disconnected session is automatically logged off after a few minutes, unless you reconnect before the logoff occurs. (There is no TSO equivalent of a VM/CMS disconnected session which allows a program to continue execution after disconnection. If you have a long running program or a program which requires many resources, you should execute it as a batch job.)

After all relevant fields have been updated, press **ENTER**. Various informational messages may appear before the logon process completes. If any lines consisting of several asterisks appear, press **ENTER** to continue the process.

3.3 TSO Logoff

Native TSO Mode

If your terminal session is in native TSO mode (identifiable by the **READY** prompt), enter the following command to logoff:

```
LOGOFF
```

It is also possible to logoff one userid and simultaneously begin the TSO logon process for another userid. This is done by entering the following command, where **newuserid** represents the userid to be used for the logon process :

```
LOGON newuserid
```

ISPF Mode

If you are using ISPF, return to native TSO mode by exiting ISPF. After exiting ISPF, follow the logoff instructions shown above for Native TSO Mode. A brief summary on how to exit ISPF is given below:

- ?? If the primary ISPF menu is not displayed, return to the primary ISPF menu. This is typically accomplished by pressing the **PF3** key repeatedly.
- ?? Once the primary ISPF menu is displayed, type **X** on the command line and press **ENTER**.
- ?? Depending on circumstances, either the **READY** prompt appears (native TSO mode) or an ISPF screen requesting the disposition of the LOG and/or LIST data sets is displayed.
- ?? If the ISPF LOG/LIST disposition screen is displayed, enter the appropriate information to either print or delete the LOG or LIST files and then press

ENTER. A batch job is submitted to print the files and then ISPF exits. (More information on LOG/LIST is given later in this document.)

?? At this point, native TSO mode will be active. Refer to the logoff instructions shown above for Native TSO Mode.

4. DATA SETS

4.1 Data Set Naming Conventions

Several data set naming conventions are used on the MVS system at ASU, the two most common being TSO and WYLBUR. These conventions are subsets of the standard IBM MVS data set naming convention. Permanent data sets which you create must comply with an appropriate naming convention.

Because the naming convention for TSO is different than the one used for WYLBUR, people often ask: Can I use my WYLBUR data sets from TSO?

The answer is **YES**. From TSO you can use data sets on the MVS system regardless of the naming convention; however, certain TSO tasks are simpler when the data sets being referenced conform to the TSO naming convention. Some TSO functions require prior decompression of data sets in WYLBUR edit format.

Several naming conventions are described below, beginning with the standard IBM data set naming convention. For data sets you create, the TSO naming convention is recommended; however, you may find the WYLBUR naming convention more convenient if you have other files which conform to the WYLBUR convention.

Standard IBM Naming Convention

- A data set name consists of one or more words called qualifiers.
- Each qualifier consists of 1 to 8 alphabetic, numeric, and/or national characters. (The alphabetic characters are the uppercase letters (**A...Z**); the numeric characters are the digits (**0...9**); the national characters are the dollar, pound, and commercial at signs (**$#@**).
- The first character of each qualifier must be either alphabetic or national.
- A period (**.**) is used as a separator between qualifiers.
- The overall length of the data set name, including periods, cannot exceed 44 characters.
- A member name enclosed in parentheses may be appended to the name of a partitioned data set if a member of the data set is being referenced. The member name consists of a one word qualifier. The member name and the enclosing parentheses do not count toward the 44 character length of the data set name.

TSO Naming Convention

- The Standard IBM Naming Convention and the conditions listed below are used.
- The data set name must have at least two qualifiers. (If it has exactly three qualifiers, it also conforms to the ISPF Naming Convention.)
- The first qualifier must be the userid of the data set owner.
- For VSAM files only, the CLUSTER, DATA, and (if present) INDEX components must explicitly be given names.

ISPF Naming Convention

- The TSO Naming Convention is used, except the data set name must have exactly three qualifiers.

WYLBUR Naming Convention

- The Standard IBM Naming Convention and the conditions listed below are used, except the national characters (**$#@**) should not be used in the data set name nor in the member name.
- The data set names must have at least four qualifiers.
- The first qualifier must be **WYL** or **VSM**. WYL is used for non-VSAM files, and VSM is used for VSAM files. (The distinction between the two types of files for naming purposes is no longer required.)
- The second qualifier must be the first two characters of the userid which owns the data set.
- The third qualifier must be the last three characters of the userid which owns the data set.
- A fourth qualifier is required; additional qualifiers beyond the fourth are optional.
- The former WYLBUR environment permitted data set name qualifiers to have more than 8 characters and to have non-standard characters. Non-standard data set names may create difficulties in other environments.
- For VSAM files only, the CLUSTER, DATA, and (if present) INDEX components must explicitly be given names.
- A member name follows the same rules as listed for Standard IBM Naming Convention, except that other characters are also allowed; however, these other characters are not recommended if the member name is to be referenced in a non-WYLBUR environment. The former WYLBUR environment permitted two methods of designating a member name: enclosing the member name in parentheses or preceding the member name with a pound sign (#); the latter is valid only in a WYLBUR environment.

Sample data set names are shown below for each naming convention. The userid ABCDE is used for those requiring a userid.

Data Set Name	Naming convention
SYS1.PROCLIB(FORV2CLG)	Standard IBM
SYS2.WYL.PUB.LIB	Standard IBM
ABCDE.CNTL(JCL)	TSO
ABCDE.SURVEY.PIMA.Y90M03	TSO
ABCDE.RURAL.COUNTY	TSO and ISPF
ABCDE.SAMPLE.DATA(PROJ2)	TSO and ISPF
WYL.AB.CDE.LIB(LOGON)	WYLBUR (non-VSAM)
WYL.AB.CDE.PROJ5.FORTRAN	WYLBUR (non-VSAM)
VSM.AB.CDE.HMWK5.INDEX	WYLBUR (VSAM)

In a batch environment (i.e., in JCL statements), data sets are referenced by fully qualified names; that is, the full name of the data set is used. If a single member of a partitioned data set is being referenced, the full name of the data set appended with the member name enclosed in parentheses must be used.

In the TSO and WYLBUR environments, under some conditions, an abbreviated form of data set names is permitted. It is always permissible to use fully qualified data set names. The table below illustrates various possibilities for referencing data set names from the TSO, WYLBUR, and batch environments.

Fully qualified data set name	Referenced from	Referenced as
ABCDE.BOTANY.PROJECT	batch JCL	ABCDE.BOTANY.PROJECT
	TSO	'ABCDE.BOTANY.PROJECT' BOTANY.PROJECT
	WYLBUR	'ABCDE.BOTANY.PROJECT' "ABCDE.BOTANY.PROJECT" $ABCDE.BOTANY.PROJECT
WYL.AB.CDE.SURVEY.DATA	batch JCL	WYL.AB.CDE.SURVEY.DATA
	TSO	'WYL.AB.CDE.SURVEY.DATA'
	WYLBUR	WYL.AB.CDE.SURVEY.DATA 'WYL.AB.CDE.SURVEY.DATA' "WYL.AB.CDE.SURVEY.DATA" $WYL.AB.CDE.SURVEY.DATA SURVEY.DATA

Though not shown above, a member name may be specified for partitioned data sets, e.g., 'WYL.AB.CDE.DATA(PROJ3)' and SAMPLE.FORTRAN(MEAN).

Note: A data set has only one name. There cannot be a TSO name for a data set and a WYLBUR name for the same data set. In TSO, a WYLBUR data set name must always

be fully specified and enclosed in single quotes. A TSO data set name is enclosed in quotes only if it is fully specified; otherwise, the TSO data set name is not quoted and the leading qualifier (the userid) is omitted.

4.2 Data Set Formats

The record format of a data set (RECFM) is the primary indicator of the data set type. The logical record length (LRECL) and blocksize (BLKSIZE) also help in determining data set type. TSO can generally work with data sets having **fixed format** (RECFM of F, FB, FBA, etc.) or **variable format** (RECFM of V, VB, VBA, etc.). Partitioned data sets having a RECFM of U and a LRECL of 0 are usually **load modules** (binary, executable form of a program) and are compatible with TSO. Partitioned or sequential data sets having a RECFM of U, a BLKSIZE of (usually) 6233 or 7476, and an LRECL equal to the BLKSIZE (but sometimes 0) may be WYLBUR **edit format** files (compressed files readable only by WYLBUR).

Data sets in WYLBUR edit format have to be converted to fixed or variable format before they can be read, written, edited, or submitted as batch jobs from TSO. The interactive TSO utility **%UNPRESS** performs the conversion.

4.3 Unpress WYLBUR Data Sets

Technically, there is no such thing as a "WYLBUR data set"; this term normally refers to a data set which conforms to the WYLBUR data set naming convention.

If a data set is not in WYLBUR edit format, no special action is needed, though you may want to rename the data set to conform with the TSO naming convention. TSO and ISPF have commands to rename data sets and members.

The TSO command for renaming a data set from the WYBUR naming convention to the TSO naming convention is shown below. (On the ISPF menu for renaming, which is not shown, you simply type the new name over the existing name.)

```
RENAME 'WYL.AB.CDE.DATA1' DATA1
```

The TSO command for renaming a member of a partitioned data set is shown below. (On the ISPF menu for renaming, which is not shown, you simply type the new name next to existing name.)

```
RENAME DATA1(OLD) DATA1(NEW)
```

If a data set is in WYLBUR edit format, the TSO utility **%UNPRESS** may be used to convert the data set. The easiest method is to use ISPF option 3.4 to display the list of WYLBUR data set names. Once the list is displayed, press PF11 one or more times until

the screen with a title line of "Dsorg Recfm Lrecl Blksz" is displayed. Use PF7 or PF8 to move the screen up or down to see other data set names. A Recfm of **U** and a Dsorg of **PO** (partitioned) or **PS** (sequential) indicates a possible WYLBUR edit format data set. (Load module libraries contain binary executable programs, have a Recfm of U and a Dsorg of PO but are not WYLBUR edit format.) (Note: No harm is done if you try to convert a load module library or other non-edit format data set.)

Tab to the line containing the name of the data set you want to convert, and type **%UNPRESS** to the left of the name and press enter. The conversion process takes several seconds to complete for a small file, longer for larger files.

You can also use **%UNPRESS** from a TSO command line. A sample TSO command for converting an edit format data set is shown below:

```
%UNPRESS 'WYL.AB.CDE.LIB'
```

Note: Some earlier versions of WYLBUR were unable to prevent you from saving a non-edit format member into an edit format library. The new member (and on rare occasions the entire library) would become unusable. UNPRESS is unable to decompress libraries with such members; however, if the offending members are deleted, UNPRESS will be successful.

5. ISPF (Menu Facility)

5.1 Overview

ISPF (Interactive System Productivity Facility) is a shell which provides a series of menus to simplify the use of TSO and provides additional functionality. Only a brief description of ISPF is given in this document.

To activate ISPF, type the following command on the TSO command line:

```
ISPF
```

Once ISPF has been started, a menu (also called a panel) is displayed. A sample primary menu for ISPF is shown below:

```
                        ISPF/PDF PRIMARY OPTION MENU
OPTION  ===>
0  ISPF PARMS    - Specify terminal and user parameters
1  BROWSE        - Display source data or output listings
2  EDIT          - Create or change source data
3  UTILITIES     - Perform utility functions
6  COMMAND       - Enter TSO command or CLIST
9  IBM SOFTWARE  - Additional IBM software
D  DATASET MGMT  - Disk/Tape Management, File Transfer
I  IOF           - Interactive Output Facility
T  TUTORIAL      - Display information about ISPF/PDF
X  EXIT          - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

The panels are used to identify the task you wish to perform. A menu panel has a list of options, each preceded by an identifier. To select an option on a menu, type the corresponding identifier (for example: **3**, **9**, or **X**) on the OPTION line and press **ENTER**. Some panels have a COMMAND line instead of an OPTION line. (The OPTION or COMMAND line can be at the top or the bottom of a panel.) This typically results in another panel being displayed. The subsequent panel may contain another menu, or it may contain one or more fields in which you need to enter required information. For example, if option **2** (EDIT) were chosen, the subsequent screen would contain input fields where you would type the name of the data set to be edited.

Note: The identifier associated with a particular item on a specific menu may change whenever ISPF or its associated menus are updated, especially if new features or options are added.

Selection of options from the main menu can also be made by typing **n.m**, where **n** is the main menu option and **m** is the sub menu option. In this manner you can move directly to menu **m** without having to first select **n** and then **m**. For example, if the main menu is not currently displayed, but you want to select **3** from the main menu and **4** from the sub menu, type **=3.4**. The equal sign (**=**) causes a jump to the main menu without displaying the menu before processing the indicated option.

You can usually return to a previous menu by pressing the **PF3** key or typing the **END** command. Pressing the **PF4** key or typing the **RETURN** command will generally return you to the main menu.

When ISPF panels are displayed, you may notice data fields contain values entered in previous ISPF sessions. In some instances you may not need to change the entries. To make changes, simply over type the field. Any changes you make may be retained and displayed the next time the same panel is displayed.

5.2 Terminal Keys

A 3270-type terminal has several programmable function keys called **PF** keys. To view the values of the PF keys or to change their values, enter the command **KEYS** on the ISPF command line. The values are usually short ISPF or editing commands. For example, the **PF3** key is usually defined as **END**; the **PF4** key is usually defined as **RETURN**.

Note: Program function keys (**PF1**, **PF2**, ..., **PF24**), program attention keys (**PA1**, **PA2**), **CLEAR** key, **INSERT** key, **RESET** key, etc., are names of keys that appear on a 3270-type terminal, which you are probably not using. The terminal emulator software running on your PC determines the keyboard mapping, i.e., determines which keyboard keys correspond to the 3270-terminal keys. Not all terminal emulators use the same mapping. If you don't know the keyboard mapping, check the documentation which came with your terminal emulator software. The **KEYS** command does not affect the keyboard mapping of your terminal emulator, and the keyboard mapping does not affect the values associated with the PF keys.

On-line help is generally available within ISPF; press the HELP key (**PF1**) to request HELP at any time. An on-line tutorial is available which explains how to use ISPF and what the various options are; select option **T** to begin the ISPF tutorial.

Panels or menus are often nested; that is, one panel may cause another panel to appear. The **PF3** key is used to exit a panel and return to the previous panel; used repeatedly, it will eventually return to the main menu. If the **PF4** key is assigned the value **RETURN**, it can generally be used to return immediately to the main menu.

The **PA1** key may be used to interrupt a process, such as an executing program. The **PA2** key is used to refresh the screen within ISPF. The **CLEAR** key is used to clear the screen; however, within ISPF, it is usually necessary to use the **PA2** key immediately after using the **CLEAR** key.

5.3 PF Key List

Some panels show a short list of available PF keys at the bottom of the screen; other panels show no list or only a partial list. To cause the list of available PF keys to always display at the bottom of a panel, enter the following command on an ISPF command line:

```
PFSHOW ON
```

You can control whether 12 or 24 PF keys are available. If you decide to have 24 PF keys, you can also control whether to have the first 12, the second 12, or all 24 PF keys appear in the menu at the bottom of the screen. Use the following ISPF command to control the number and format of PF keys displayed:

```
PFSHOW TAILOR
```

5.4 PF Key Values

The value of PF keys may vary from one panel to the next; i.e., there are multiple sets of PF key definitions, each set having a unique name. One group of panels may use a particular set of PK key definitions, and another group may use the same or a different set of definitions.

You can change the value of the PF keys; however, it is recommended you become familiar with the existing values and identify which panels use them before making any changes.

Several methods exist for making changes. Use ISPF option 0.3 to change the main set of PF key definitions. Use ISPF option 0.8 to change other sets of PF key definitions, provided you know the name of the set to be changed. If you don't know the name, go to the panel for which the values should be changed and enter the ISPF command shown below.

```
KEYS
```

5.5 Help Key

On most ISPF panels, the value of the **PF1** key is **HELP**. You can either press the PF1 key or enter the command HELP to get context sensitive help information, which is especially useful after receiving an error message.

If you request help (by pressing PF1 or entering HELP) immediately after receiving an error message, ISPF displays additional information regarding the error which has just occurred. Requesting help again may display more information on the same subject or on a broader scale.

5.6 Technical Notes

ISPF requires a partitioned data set be allocated to the ddname ISPPROF; the allocation is done automatically at logon time. At ASU, the data set is named 'userid.ISPFLIB', where "userid" is replaced with your userid. This data set (usually two tracks) should not be deleted; ISPF uses it for processing and to retain certain values from one terminal session to the next. The data set is created automatically when you logon TSO the first time. Should you ever accidentally delete the data set, it will be created again during the next logon.

ISPF requires a LOG file and a LIST file (both are print files) allocated to the ddnames ISPLOG and ISPLIST, respectively. This is done automatically. You can use ISPF option 0.2 to exercise additional control of these data sets, such as setting the default **Process option** for the LOG file to **D** (delete) and for the LIST file to **PD** (print, then delete). You can also use the JCL template at the bottom of the LOG/LIST disposition screen to establish a fully specified JOB statement and a **/*ROUTE PRINT** statement. (Do not use the line labeled **Local printer ID**.) The JOB statement and the **/*ROUTE PRINT** statement are used to submit a batch job which prints the files each time you exit ISPF. (The WYLBUR form of a JOB statement is not valid in TSO.)

The LOG file contains a log of your current terminal sessions, and the LIST file contains the printed results of certain ISPF commands. (Printed output from batch jobs are not part of these two files.) These files use disk space which counts toward your disk quota. ISPF option **0.2** includes fields for determining the size of these files

6. EDITOR

The editor provided with ISPF/PDF is ISREDIT. It is commonly referred to by a variety of names, including SPF, PDF, SPF/PDF, ISPF/PDF, or ISREDIT. It is a full-screen editor and may be used to edit a member of a partitioned data set (PDS) or a sequential data set. ("Library" is another common term which refers to a PDS.) Only a brief description of the editor is given in this document.

You can type new text or type over existing text anywhere on the screen. Certain editing commands, such as **I**, **C**, **M** or **D** (Insert, Copy, Move or Delete), can be entered in the line number fields. Other editing commands are entered on the command line. Use the **PF3** key to end editing and automatically save the changes, or use **CANCEL** to exit without saving the changes. **SAVE** may also be used periodically to save the changes without exiting the editor.

If the file being edited is intended for batch processing, the editor's **SUBMIT** command may be used to submit the displayed file to the batch queue. IOF may be used to monitor the batch job or to retrieve its output.

The SPF/PDF editor is invoked from the ISPF Utilities/Dslist menu (option **3.4**) or from the primary ISPF menu (option **2**).

ISPF option **3.4** is the easiest method of invoking the editor. This option displays a list of data set names. When the list is displayed, tab to line containing the name of the data set to be edited, type an **E**, and press enter. (When using the editor via option 3.4, an abbreviated Edit Entry Panel will sometimes appear. If it does, just press enter to continue.)

ISPF option **2** is more involved and is described in the following paragraphs. (When using the editor via option 2, the entire Edit Entry Panel will always appear.)

When option **2** is selected, the editor displays the Edit Entry Panel containing several input fields, including a field for the name of the data set to be edited. After you have typed the data set name and pressed enter, the editor will display the file if it contains data or will display an input screen if the file is empty. A sample edit entry panel is shown below:

```
                         EDIT ENTRY PANEL
COMMAND ===>

ISPF LIBRARY:
   PROJECT ===>
   GROUP   ===>
   TYPE    ===>
   MEMBER  ===>      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
   DATA SET NAME  ===>
   VOLUME SERIAL  ===>     (If not cataloged)

INITIAL MACRO     ===>            Enter "/" to select option
PROFILE NAME      ===>            / Confirm Cancel/Move/Replace
FORMAT NAME       ===>            _ Mixed Mode

DATA SET PASSWORD ===>     (If password protected)
```

The data set name may be entered in either of two locations: the DATA SET NAME field or the ISPF LIBRARY fields (PROJECT, GROUP, and TYPE). If both the DATA SET NAME field and the ISPF LIBRARY fields contain data set names, the DATA SET NAME field is used. One useful feature of the ISPF LIBRARY fields is the data set name is retained from one terminal session to the next, but not for the DATA SET NAME field.

The DATA SET NAME field is below "OTHER ... DATA SET" in the middle of the screen. This field may be used for any sequential or partitioned data set name. If the data set is a PDS, a parenthesized member name should be appended to the data set name. The data set name and any member name should be enclosed within single quote marks. If the first qualifier of the data set name matches your PREFIX, the first qualifier and the quote marks may be omitted. By default your PREFIX is equivalent to your userid, unless you use the TSO PROFILE command to change the PREFIX.

The member name is required for creating a new member and is optional for editing an existing member. If the member name is omitted for a PDS, a menu of existing members is displayed; tab to the name of the member to be edited, type an **S** next to the member to be edited and press enter.

Either the ISPF LIBRARY fields or the DATA SET NAME field may be used if the fully qualified data set name contains exactly three qualifiers. Refer to the previous paragraph for instructions on how to use the DATA SET NAME field. For the ISPF

LIBRARY fields, enter the first qualifier of the data set name (typically your userid) in the PROJECT field; enter the second qualifier in the first GROUP field; enter the last qualifier on the TYPE field. If the data set is a PDS, enter a member name in the MEMBER field; the member name is optional for editing an existing member. If the member name is omitted for a PDS, a menu of existing members is displayed; tab to the name of the member to be edited, type an **S** next to the member to be edited and press enter.

The next example illustrates selecting a member from a menu of existing members:

```
EDIT --- ABCDE.MOON.DATA7 ---------------------- ROW 00001 OF 00004
COMMAND ===>                                       SCROL L ===> PAGE
    NAME          VV.MM  CREATED      CHANGED     SIZE  INIT  MOD    ID
   ARMSTRON       01.00 91/08/07 91/08/07 17:43    2     2     0   ABCDE
   CHEESE         01.00 91/08/07 91/08/07 17:26    2     2     0   ABCDE
s  CRATER         01.00 91/08/07 91/08/07 17:13    1     1     0   ABCDE
   MOUNTAIN       01.00 91/08/07 91/08/07 17:14    1     1     0   ABCDE
   **END**
```

Before the editor can be used to enter data into a new data set, the data set must exist as an empty file. ISPF option **3.2** or the TSO command **ALLOCATE** may be used to create a new empty data set.

Regardless of whether you are using ISPF option **2** or **3.4**, there are five possible editing situations:

- ?? Edit an existing sequential data set.
- ?? Edit a new sequential data set; an empty sequential data set must be created prior to editing.
- ?? Edit an existing member in an existing partitioned data set.
- ?? Edit a new member in an existing partitioned data set.
- ?? Edit a new member in a new partitioned data set; an empty partitioned data set must be created prior to editing.

In only two situations is it necessary to create a new, empty data set prior to invoking the editor. Also, note the difference between editing a new member of an existing data set and editing a new member of a new data set; only the latter requires a new, empty data set to be created. (ISPF option **2** is often easier than **3.4** for creating a member in an empty partitioned data set.)

As a general rule, it is preferable to create a new file as a member of a partitioned data set than to create it as a sequential data set. Various editing tasks are simpler when working with partitioned data sets; and some commands, such as copying specific lines from one file to another, are not possible with sequential data sets but are with

partitioned data sets. Sequential data sets utilize disk space inefficiently for small files and are more cumbersome to create.

7. BATCH JOB JCL

Fully specified JCL (Job Control Language) is required for batch jobs submitted from TSO. (From WYLBUR, either fully specified JCL or partial JCL is required.) The term "fully specified JCL" refers primarily to the following JCL statements:

JCL Statement	Usage note
JOB	required
ROUTE	optional
JOBPARM	optional
TAPES	optional
OUTPUT	optional
JCLLIB	optional

The above statements are in addition to any other JCL statements normally used in a batch job. The other statements, primarily EXEC and DD statements, are those required by the software (SAS, FORTRAN, etc.) being executed in the batch job.

The **SUBMIT** command is used to submit the batch job for processing. The output from the batch job may be printed, purged, or kept in the queue to be viewed at the terminal. JCL statements determine whether the output is printed or kept in the queue for viewing. The IOF facility is used to view the output, change print destinations, purge output, etc. The edit macro %MAKEJOB may be used to create a fully specified JOB statement.

The following illustrates proper syntax of the aforementioned JCL statements:

```
//ABCDE### JOB (12345),NOTIFY=ABCDE,MSGCLASS=A,REGION=8M,TIME=(1,30)
/*ROUTE PRINT KEEP              <== usually not needed
/*JOBPARM LINES=10              <== usually not needed
/*TAPES 1                       <== usually not needed
//BANNER OUTPUT DEFAULT=YES,JESDS=ALL,ROOM=#C2  <== usually not needed
// JCLLIB ORDER=SOME.PROC.LIB    <== usually not needed
```

To modify the above for your batch jobs, you would do the following:

- ?? Replace ABCDE (both occurrences) with your userid.
- ?? Replace ### with zero to three alphanumeric characters (numeric recommended).
- ?? Replace 12345 with your account number (not your userid).
- ?? Replace REGION value with another value, if necessary.
- ?? Replace TIME value with another value, if necessary.
- ?? Replace KEEP on the ROUTE statement with another print destination, if necessary. If your default print destination is KEEP, this statement is usually omitted.
- ?? Replace LINES value on the JOBPARM statement with a larger value, if necessary. If your print requirements are less then 10 thousand lines, this statement is usually omitted.

?? Replace the TAPES value with number of needed concurrent tape drives (not tape volumes). If you are not using tapes, this statement is omitted.
?? Replace the ROOM value on the OUTPUT statement if you have an assigned security bin. This statement is usually omitted.
?? Replace the procedure library name SOME.PROC.LIB on the JCLLIB statement with the name of the procedure library to be used. This statement is usually omitted.

7.1 JOB Statement

The **JOB** statement is required; however, many of the parameters (positional and keyword) may be omitted. The only required items are the jobname and the account number. In the example shown below ABCDE### is the jobname and 12345 is the account number.

```
//ABCDE### JOB (12345),NOTIFY=ABCDE,REGION=4M,TIME=(0,5)
```

An **alternative method** for specifying a JOB statement is available, though limited in functionality. This method requires the JOB statement to be replaced in its entirety by a JCL comment statement. A JCL comment statement contains **//*** in columns 1-3 and a blank in column 4. With this method, default values are chosen for all parameters. A jobname is generated consisting of your userid and up to three additional characters; the system prompts for the additional characters.

Note: The edit macro **%MAKEJOB** is available in an editing session. When entered on the editor command line, %MAKEJOB inserts a fully specified JOB statement into the file being edited. After the JOB statement is inserted, you can adjust the REGION and TIME values if necessary.

The **jobname** (required) begins in column 3, is 5 to 8 characters in length, and is followed by at least one blank. The first part of the jobname must be the userid. The remaining characters, if any, may be alphanumeric (**A**..**Z**, **0**..**9**) or national (**@#$**) characters.

When a batch job is submitted and the jobname consists of the userid only, the system prompts for additional jobname characters. The JOB statement should have enough space after the jobname (or at the end of the first line of the JOB statement) to allow insertion of additional jobname characters, else truncation of the jobname will occur.)

7.1.1 Positional Parameters

Positional parameters appear on the JOB statement after the word **JOB**, before any keyword parameters, and must appear in a specific sequence.

The **accounting** parameter (required) is the first positional parameter on the JOB statement. The **account number** (required) is the first value in the accounting

parameter. The account number must be valid for the userid specified in the jobname. A **room code** may appear after the account number. (Historically, the room code in the JOB statement was used as a plot distribution code at ASU.) A comma is used to separate the account number from the room code. Parentheses are used to enclose the accounting parameter. If only the account number is present, the enclosing parentheses are optional.

The **name** parameter (optional, not shown) is the second positional parameter. It is enclosed within single quote marks and may contain up to 20 characters of identifying information, such as the programmer's name.

7.1.2 Keyword Parameters

All keyword parameters (identifiable by an equal sign) may be specified in any order on the JOB statement and must appear after all positional parameters. There are several keyword parameters, all of which are optional. Only the commonly used parameters are described in this document.

The **NOTIFY** keyword parameter (optional) identifies a userid to which a job-completion message should be sent once the batch job completes. It is not necessary for the NOTIFY userid to match the jobname userid, though this is usually the case. (The WYLBUR equivalent, **`// EXEC WNOTIFY`**, should not be used.)

The **REGION** keyword parameter (optional) declares the amount of virtual memory needed by the batch job. If the system default is sufficient, this operand may be omitted.

The **TIME** keyword parameter (optional) declares the amount of CPU time needed by the batch job. If the system default is sufficient, this operand may be omitted.

The **MSGCLASS** keyword parameter (optional) defines the print class for the JCL listing and messages. **`MSGCLASS=Z`** may be used to discard the JCL listing and messages; however, this is not recommended since many error messages would be discarded, too. If the default MSGCLASS for your userid is sufficient, this operand may be omitted.

You may display or change the default MSGCLASS. The following **LISTUSER** command displays the default MSGCLASS for userid ABCDE. The following **ALTUSER** command changes the default MSGCLASS to A. These commands are TSO/RACF commands; the TSO command prefix is required if the commands are entered on a non-TSO command line.

```
COMMAND ===> TSO LISTUSER ABCDE NORACF TSO
COMMAND ===> TSO ALTUSER ABCDE TSO(MSGCLASS(A))
```

The **MSGLEVEL** keyword parameter (optional, not shown) determines the amount of job control information (JCL, messages) to be displayed in the job listing. If the system default is sufficient, this operand may be omitted. The parameter value contains 2 numbers separated by a comma and is enclosed in parentheses, for example, **MSGLEVEL=(2,1)**. The first number applies to statements, the second to messages.

```
Statements:
     0       Only JOB statements
     1       All JCL and procedure statements
     2       Only JCL statements
 Messages:
     0       Only JCL messages
     1       JCL, JES, and operator messages
```

Refer to the IBM JCL Reference manual for information regarding other keyword parameters and for additional information on parameters previously mentioned.

7.2 ROUTE Statement

The **ROUTE** statement may be omitted if the default print job destination for your userid is sufficient. (Use the **LISTUSER** command to display the default print destination for your userid; use the **ALTUSER** command to establish a different default print destination.) The print destination **KEEP** is recommended. It is similar to the WYLBUR FETCH queue; i.e., the batch job output is kept in the print queue for viewing. (IOF may be used to view the batch job output and to print or purge it.)

The following **LISTUSER** command displays the default print destination (and several other items) for userid ABCDE. The following **ALTUSER** command changes the default print destination to KEEP (and may be used to change other items, too). These commands are TSO/RACF commands; the TSO command prefix is required if the commands are entered on a non-TSO command line.

```
COMMAND ===> TSO LISTUSER ABCDE NORACF TSO
COMMAND ===> TSO ALTUSER ABCDE TSO(DEST(KEEP))
```

7.3 JOBPARM Statement

The **JOBPARM** statement sets the maximum number of print lines and may be omitted if the default LINES value is sufficient. **LINES=10** is the default, which represents 10 thousand print lines.

```
/*JOBPARM LINES=10
```

7.4 TAPES Statement

The TAPES statement is used to identify the number of concurrent tape drives (not tape volumes) needed by your batch job. The maximum is 4. If you are not using tapes, you should omit this statement.

```
/*TAPES 1
```

7.5 OUTPUT Statement

Printed output from batch jobs are sorted alphabetically by the last letter of the userid and placed on shelves in the computing sites, unless a security bin number appears on the banner page. If you have an assigned security bin, use the OUTPUT statement to identify the security bin number. (The OUTPUT statement appears after the JOB statement.) Any printed output with a bin number will be shelved in the corresponding security bin. In the following example, the security bin is C2; replace C2 with your assigned security bin number.

```
//BANNER OUTPUT DEFAULT=YES,JESDS=ALL,ROOM=#C2
```

7.6 JCLLIB Statement

The JCLLIB statement identifies one or more procedure libraries to be searched in addition to the standard system procedure libraries. When specifying more than one procedure library, place a comma between each library name and enclose the entire list in parentheses.

Procedure libraries listed on the JCLLIB statement are searched in the order they appear on the statement and are searched before the standard system libraries. The search order for standard system libraries is SYS1.ASU.PROCLIB and SYS1.PROCLIB. (SYS1.ASU.PROCLIB becomes part of the standard system search order effective Dec. 21, 1998.)

```
//  JCLLIB  ORDER=SOME.PROCLIB
//  JCLLIB  ORDER=(MAIN.PROC.LIB,ANOTHER.LIB.NAME)
```

8. BATCH JOB SUBMITTAL

The SUBMIT command is used to submit batch jobs from TSO to the MVS batch system for processing. The procedure is nearly identical to that used by WYLBUR; the primary difference being TSO requires fully specified JCL.

The output from the batch job may be printed, purged, or kept in the queue to be viewed at the terminal. JCL statements determine whether the batch job output is

printed or kept in the queue for viewing. The IOF facility is used to monitor batch jobs, view the output, change print destinations, purge output, etc.

There are actually two forms of the SUBMIT command: the native TSO command and the SPF/PDF editor sub-command.

Command type	Syntax
SPF/PDF editor sub-command	SUBMIT
native TSO command	SUBMIT dsname SUBMIT (dsname dsname ...)

8.1 SPF/PDF SUBMIT Command

The SPF/PDF SUBMIT command is used only when the editor is displaying a file whose contents is the batch job to be submitted. In other words, use the editor to display the file you want to submit for batch processing.

Once the file to be submitted is displayed on the screen (and after you have made any desired changes), enter the SUBMIT command on the editor command line:

```
COMMAND ===> SUBMIT
```

The entire file is submitted for processing, unless a line range option is specified to cause only a portion of the file to be submitted.

8.2 TSO SUBMIT Command

The TSO SUBMIT command may be used at any time, even when the editor is being used. The operand for this SUBMIT command is a parenthesized list containing the names of the data sets to be submitted for batch processing. If only one data set name is present, the enclosing parentheses may be omitted. Any non-quoted data set name is assumed to have "CNTL" as the last qualifier; however, if no such data set exists then the data set name without "CNTL" is used.

For example, assuming the userid is ABCDE and the job to be submitted is the member MYJOB in data set ABCDE.TSOLIB.CNTL, the command to submit the job is:

```
COMMAND ===> TSO SUBMIT TSOLIB(MYJOB)
```

Note: TSO first searches for **TSOLIB.CNTL(MYJOB)** and then for **TSOLIB(MYJOB)**. TSO submits the first one found.

Note the word **TSO** in the above command. This word is required as a command prefix for any TSO command entered on an ISPF command line or an SPF/PDF editor

command line. If a TSO command is entered on a TSO command line or on the COMMAND screen of ISPF (option **6** on the primary menu), the command prefix is not required.

A common instance of when to use the native TSO command with multiple data set names is when the body of the batch job is contained in one file but the JCL JOB statement (and perhaps other JCL statements) are in a separate file.

```
COMMAND ===> TSO SUBMIT (TSOLIB(JCL) TSOLIB(PROG1))
```

Note the use of parentheses enclosing the data set names and the use of the TSO command prefix.

A common mistake made with the TSO SUBMIT command occurs when one of the files being submitted is also being edited. Unlike, the SPF/PDF SUBMIT command, the TSO SUBMIT command uses the contents of the file as it currently exists on disk. Any editing changes made to the file do not participate in the TSO SUBMIT unless the updated file is saved prior to submission.

Besides syntax, there is another difference between the editor SUBMIT command and the TSO SUBMIT command. The TSO SUBMIT requires the data set(s) to have card-image format (i.e., RECFM FB and LRECL 80); the editor SUBMIT allows fixed or variable format (RECFM FB or VB) and any logical record length. (Actually, the editor command copies the displayed data set into a temporary data set having card-image format and then uses the TSO command to submit the temporary data set for processing.)

For either SUBMIT command, if the **jobname** of a batch job consists of the userid only, the system prompts for additional jobname characters. One or more numbers are recommended as the additional characters. (The maximum size jobname is eight characters. If the userid is five characters then one to three additional characters should be typed at the prompt.)

9. IOF (Fetching Batch Job Output)

The facility which provides monitoring and control of MVS batch jobs is called IOF (Interactive Output Facility). IOF provides facilities for monitoring the progress of batch jobs queued for execution, executing, and completed. IOF can also be used to cancel active jobs and to review, print, purge, or redirect output from jobs that have completed. IOF performs the same functions as the WYLBUR commands CANCEL, FETCH, LOCATE, PRINT, ROUTE, and PURGE.

9.1 Invoking IOF

IOF is invoked by choosing option I from the ISPF main menu. Once selected, a screen similar to the following appears. (The first time you use IOF a one-time screen will appear which tells you how to customize IOF. Press Enter to continue.)

```
                        IOF JOB LIST MENU
COMMAND ===>                                          SCROLL ===> PAGE
------------------------ RUNNING JOBS -------------------------------
---JOBNAME---JOBID--ACT-STA-SYID----CPU---I/O---STEP-----PROCSTEP-SWP
_ 1 ABCDE    T04982         MVS2    :02   8.79  ISPFPDFS           IN
------------------------- OUTPUT JOBS -------------------------------
---JOBNAME---JOBID--ACT-STA-OWNDER---DEST/DEVICE---RECS--------------
_ 2 ABCDE36  J05240         ABCDE    UCC1            937
_ 3 ABCDE41  J05311         ABCDE    KEEP           2307

F1=HELP   F2=SPLIT    F3=END      F4=RETURN   F5=IFIND    F6=FIND ' -
F7=UP     F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

To look at the batch job output, tab to the job of interest and type **B** to the left of the jobname and press enter.

To print the batch job, tab to the job of interest and thence to the destination field; type a printer name over the existing destination and press enter.

To exit IOF, press the END key (**PF3**) one or more times.

To get the IOF main menu, type **/** on the command line and press **ENTER**. A portion of the main menu is shown below.

```
                        IOF USER OPTION MENU
COMMAND ===>

Select an option.

blank = Your jobs     G = Output groups           QT = Quick tutorial
  I   = Input jobs    J = All jobs menu         HELP = Help
  R   = Running jobs  H = Jobs with held output
  O   = Output jobs   P = Profile

JOBNAMES ===>
```

9.2 On-line Help and Tutorial

On-line Help is available within IOF. Type **HELP** on any command line within IOF or press the HELP key (**PF1**), and a help screen will appear. Invoking Help from the IOF

main menu displays a general IOF help menu; invoking Help while using a specific function displays Help specific to that function.

An on-line tutorial dealing with all the major functions of IOF is available from any screen within IOF. Type **QT** from any command line within IOF and follow the instructions given.

9.3 Running Jobs (Batch Jobs)

Running jobs are those jobs currently executing, including your current TSO session. Use the prefix command **B** to browse (view) the output of a running job or **C** to cancel it. The command is typed in the leftmost position of the IOF line corresponding to the job to be acted upon. (When browsing a job that has begun running but its output is not yet available for viewing, the output displayed may be similar to that of a job which has canceled.) Refer to the IOF on-line Help and tutorial for information on other commands.

9.4 Input Jobs (Batch Jobs)

Input jobs are those jobs awaiting execution. Use the prefix command **B** to browse (view) the job or **C** to cancel it. The command is typed in the leftmost position of the IOF line corresponding to the job to be acted upon. Refer to the IOF on-line Help and tutorial for information on other commands.

9.5 Output Jobs (Print Jobs)

Output jobs are those jobs which have completed execution. If a job's print destination is KEEP, it will remain in the mainframe print queue for browsing; otherwise, it is printed. Use the prefix command **B** to browse (view) or **C** to cancel (purge) the job or **S** to enter the IOF Job Summary panel. The command is typed in the leftmost position of the IOF line corresponding to the job to be acted upon.

The S prefix command displays the IOF Job Summary panel, which lists a menu of individual print files within the batch job. The B (Browse) or **P?** (Print) prefix commands can be used on the individual files instead of on the entire job. Also, from within the job summary panel, the entire job or selected print files can be saved to disk. See the "Save Batch Output to Disk" section for more information.

To print an entire job (not recommended), type a valid print destination over the existing one on the IOF main panel or on the IOF Job Summary panel. To print selected print files within the batch job (recommended), use the P? prefix command on the IOF Job Summary panel for the desired print files. P? displays another panel containing a print destination field; type a valid destination on this latter panel to print the selected

file. (By default, printing the entire job results in the output job being removed from the mainframe print queue. Printing selected portions with P? does not remove the output job from the mainframe print queue.)

Note: If the print destination is an ASU IT-provided printer in a public site, the print job is not physically printed until after payment is made for the printed output. Payment must be made within two hours of initiating the print request else the print request is purged from the printer's queue. If the print request is purged, another print request can be initiated provided the output job still exists in the mainframe print queue. (The P? method is recommended for printing files because P? does not remove output jobs from the mainframe print queue.)

Refer to the IOF on-line Help and tutorial for information on other commands.

9.6 Held Output

Any print files using SYSOUT class W are considered Held Output. (SYSOUT class W is the standard TSO Hold Output print class at ASU.) Print files using print classes other than W appear in Output Jobs. Use of SYSOUT class W is an alternative method of keeping output in the print queue for subsequent browsing. Use of print destination KEEP, which accomplishes the same thing, is recommended since it allows all print files to appear on the same display screen.

To print a held output file, type a valid class letter (typically **A**) over the existing class letter and follow the procedure listed in the previous section for Output Jobs. Refer to the IOF on-line Help and tutorial for information on other commands.

9.7 Save Batch Output to Disk

Some portion or all of a completed batch job may be saved to disk. On the IOF Job List Menu, enter the prefix command S in the leftmost position of the IOF line corresponding to the job to be acted upon. This causes the IOF Job Summary screen to be displayed.

To save the entire job to disk, enter the following three primary commands on the command line. (Read this entire section before using SD.)

```
SD         /* Display menu whereon the output disk file is chosen */
SNAP       /* Save the entire batch job to disk */
SNAPCLOS   /* Close the disk file */
```

The process to save one or more print files to disk is similar to saving the entire job to disk, except the **SNAP** primary command is replaced by one or more **N** prefix commands. The N prefix command is typed in the leftmost position of the IOF line

corresponding to the print file to be acted upon. The following sample saves three print files to disk. (Read this entire section before using SD.)

```
SD         /* Display menu whereon the output disk file is chosen */
N          /* Prefix command to save print file to disk */
N          /* Prefix command to save print file to disk */
N          /* Prefix command to save print file to disk */
SNAPCLOS   /* Close the disk file */
```

The SD command with no operands causes the IOF Snap Data Set Options screen to be displayed whereon you choose the disk file into which the output is to be saved. The default data set name is **`jobname.IOFLIST`**. You can choose another name or change other settings associated with the data set. The data set will be automatically created if it does not already exist.

The SD command with the DATA operand, which identifies an existing data set for saving the output, bypasses display of the IOF Snap Data Set Options screen. The previous contents of the data set are overwritten. The syntax is as follows:

```
SD DATA(datasetname)  /* Identify output disk file */
```

10. RACF (Disk and Tape Security)

RACF (Resource Access Control Facility) is used to define account attributes, provide data set security, and control access to various system components. Security is based on userids.

RACF can be accessed via ISPF option 9 (IBM SOFTWARE) or as commands on the TSO or ISPF command line. If accessed via ISPF, a RACF tutorial is available; select option T on the RACF main menu. (RACF commands entered on an ISPF command line should begin with the word **TSO**.)

From the RACF menu in ISPF, option **1** is used to change disk file security, option **2** is used for tape volume security, and option **4** is used to display or change various attributes for your account. Refer to the on-line help and tutorial for information on how to use the RACF panels.

10.1 Profile Names

Information contained in the RACF data base defines the security for everything within the system. Each entry in the data base is called a profile, and each entry has a name. Typically, a data set name (or a pattern which matches a data set name) or a tape volume serial number is used as the name for a RACF profile.

The RACF commands and menus assume that data set names conform to the TSO naming convention. Any data set names which do not conform to the TSO naming convention or for which the first qualifier does not match your PREFIX (typically your userid) should be enclosed in quotes.

In its standard configuration, RACF does not work with non-TSO naming conventions. At ASU, RACF internally gives TSO-type names to RACF profiles associated with WYLBUR-type names. For example, if a RACF profile were created for the data set 'WYL.AB.CDE.DATA', RACF would create a profile named 'ABCDE.WYL.DATA. Generally, you need not be aware of the difference; however, when displaying a list of existing profiles, the RACF generated names are be displayed instead of the WYLBUR-type names.

10.2 Disk File Security

You can give one or more userids access to one or more disk data sets. The data set(s) may be identified by a specific name (one data set) or by a pattern containing asterisks (many data sets). The data set name or pattern is also the RACF profile entry name. For pattern matching, a single asterisk (*) matches one qualifier; double asterisks(**) match multiple qualifiers.

```
'ABCDE.PROJECT.DATA'    /* matches one data set */
'ABCDE.*.SAS'           /* many data sets with three qualifiers */
'ABCDE.PROJECT.*'       /* many data sets  with three qualifiers */
'WYL.AB.CDE.**'         /* all data sets with WYLBUR-type names */
'ABCDE.**'              /* all data sets */
```

Note: Because RACF internally gives TSO-type names to RACF profiles associated with WYLBUR-type names, the last example given above ('ABCDE.**') would match all data set names, both TSO- and WYLBUR-type names.

10.3 Tape Volume Security

Tape security is applied to the entire tape rather than to individual files. The name of a tape RACF profile is the tape's volume serial number. (Note: Tape profiles are known as "general resources" and belong to the "tapevol" class. This information is needed when using the ISPF menus for changing security associated with tape volumes.)

10.4 LISTUSER

The LISTUSER command displays two separate components, one for RACF and the other for TSO. The command can display either or both components.

```
LISTUSER userid                  /* list RACF information only    */
LISTUSER userid NORACF TSO       /* list TSO information only     */
LISTUSER userid TSO              /* list RACF and TSO information */
```

10.4.1 Display Disk Quota

A sample listing of the first few lines of the RACF information is shown below. Disk quota information is shown on the INSTALLATION-DATA line. The number associated with MAX is the userid's maximum disk space quota in tracks; the number associated with USED is the number of tracks currently used by the userid.

```
USER=?????  NAME=?????????          OWNER=?????     CREATED=89.140
DEFAULT-GROUP=USERSGRP0 PASSDATE=91.190  PASS-INTERVAL= 90
ATTRIBUTES=NONE
REVOKE DATE=NONE   RESUME DATE=NONE
LAST-ACCESS=91.247/18:55:48
CLASS AUTHORIZATIONS=NONE
INSTALLATION-DATA=MAX(00100) USED(00000025)
NO-MODEL-NAME
LOGON ALLOWED   (DAYS)          (TIME)
-----------------------------------------
ANYDAY                          ANYTIME
```

Note: As of April 2002, disk quotas are not enforced. The information has been left in this document for historical purposes.

10.4.2 Display Account Number

A sample listing of the TSO information is shown below. The items of note are the default MSGCLASS for batch jobs (MSGCLASS), the default SYSOUT class for printed output (SYSOUTCLASS), the default batch job print destination (DEST), the account number (ACCTNUM), the default disk unit for creating new disk data sets (UNIT), the amount of virtual memory being used (SIZE), and the maximum amount of virtual memory available for the terminal session.

```
TSO INFORMATION
ACCTNUM=12345
DEST=UCC1
HOLDCLASS=W
MSGCLASS=A
PROC=ISPFPDFS
SIZE=0004096
MAXSIZE=0008192
SYSOUTCLASS=A
UNIT=RESRCH
```

10.5 ALTUSER

The ALTUSER command may be used to change certain items (but not all) in the RACF and TSO components displayed by the LISTUSER command. Several of the more useful commands are shown below:

```
ALTUSER userid TSO(DEST(KEEP))    /* Set print dest to "fetch" queue */
ALTUSER userid TSO(MSGCLASS(A))   /* Set MSGCLASS for batch */
```

There is no need to change PROC or SIZE since these two will not take effect until the next logon, at which point you can alter them on the logon screen.

11. TSO COMMANDS

TSO "commands" may be built-in commands, programs, or execs. Regardless of the type, the commands have the same general format and are entered on the TSO command line. The commands may also be typed on the ISPF command line but must have the word **TSO** appended to the front of the command; this word informs ISPF that the command should be submitted to TSO for processing.

(ISPF also has a set of commands. These commands can only be entered on an ISPF command line, not on a TSO command line.)

11.1 ALLOCATE

The ALLOCATE (or ALLOC) command is used to allocate files. It performs the same function as a JCL DD statement. It associates a ddname (DD) with a data set name (DSN). With some software the ALLOC command is optional. For example, SAS and FORTRAN are able to read or write files with or without using the ALLOC command; other software need the ALLOC command.

Data set names in the ALLOC statement may be fully qualified, and, if the data set is partitioned, a parenthesized member name may be appended to the data set name. Single quote marks should enclose fully qualified data set names. Without the enclosing quote marks, the system will assume that the current PREFIX (typically your userid) should be affixed to the data name to form a fully qualified data set name. For input files, a list of data set names may be used to logically concatenate the files into one larger file.

```
Write to an existing data set:
      ALLOC DD(ddname) DSN(datasetname) OLD

Read from an existing data set:
      ALLOC DD(ddname) DSN(datasetname) SHR REUSE

Read from an existing data set (used by FORTRAN programs):
      ALLOC DD(ddname) DSN(datasetname) SHR REUSE INPUT

Write to a new sequential data set:
      ALLOC DD(ddname) DSN(datasetname) NEW CATALOG REUSE
        SPACE(primary secondary) TRACKS UNIT(RESRCH) RELEASE
        RECFM(F B) LRECL(80) BLKSIZE(7440)
```

```
Write to a new partitioned data set:
      ALLOC DD(ddname) DSN(datasetname(member)) NEW CATALOG REUSE
        SPACE(primary secondary) TRACKS DIR(blocks) UNIT(RESRCH)
        RECFM(V B) LRECL(259) BLKSIZE(7476)

Write to the printer:
      ALLOC DD(ddname) SYSOUT(A) DEST(destination)

Read from or write to the terminal:
      ALLOC DD(ddname) DSN(*)

Read from or write to a "dummy" file:
      ALLOC DD(ddname) DUMMY
```

?? ALLOC is the command.
?? DD(...) identifies the ddname.
?? DSN(...) identifies the data set name(s) and/or member(s). DSN(*) represents the terminal.
?? OLD indicates the data set exists and will be read and/or written.
?? SHR indicates the data set exists and will be read.
?? INPUT is used along with SHR but is only needed by FORTRAN programs.
?? NEW indicates the data set does not yet exist but is being created.
?? CATALOG is used along with NEW to catalog a new data set.
?? REUSE indicates that if the ddname is already in use, the existing allocation should be freed and the ddname reused.
?? SPACE indicates the size of a new data set; "primary" is the initial size and "secondary" is the increment needed if the primary size is too small.
?? TRACKS indicates that SPACE is allocated in tracks. CYL is used to allocate space in cylinders.
?? DIR indicates that a partitioned data set is being created and "blocks" indicates the number of directory blocks needed.
?? RELEASE returns unused tracks or cylinders to the system.
?? UNIT indicates the disk volumes to contain new data sets.
?? RECFM is the record format.
?? LRECL is the logical record length.
?? BLKSIZE is the blocksize of the data set.
?? SYSOUT indicates a printer file.
?? DEST indicates the print destination.
?? DUMMY indicates the file is a dummy file.

11.2 CALL

CALL is used to execute a load module, the binary executable form of a program which has previously been link-edited. If the first qualifier of the data set name containing the load module does not match your PREFIX (typically your userid) or if the last qualifier is not LOAD, the data set name must be fully qualified and enclosed within single

quotes on the command line; otherwise, both the first and last qualifier should be omitted on the command line.

```
Execute a load module and pass runtime parameters:
      CALL 'fulldatasetname(member)' 'parameters'

Execute a load module with a shortened name:
      CALL shortname(member)
```

11.3 DELETE

DELETE is used to delete a data set. ISPF also has several methods to delete a data set.

```
Delete a member:
      DELETE datasetname(member)

Delete a data set:
      DELETE datasetname
```

11.4 EXEC

11.4.1 Explicit Execs

EXEC is used to explicitly execute a CLIST or REXX exec. (CLISTs and REXX execs may also be executed implicitly if the libraries containing the CLISTs and REXX execs are assigned to the ddnames SYSPROC or SYSEXEC.)

You may create your own execs using either REXX or CLIST, but REXX is recommended. It is highly recommended that the first line of any REXX exec begin with the comment **`/*REXX*/`**. Depending on the method used to invoke the exec, the comment is sometimes required.

The EXEC command requires a data set name (including a member name if necessary). If the data set name is fully qualified or if the first qualifier of the data set name does not match your PREFIX (typically your userid) or if the last qualifier of the data set name is not CLIST or EXEC, the data set name must be enclosed in parenthesis.

If the first data set name qualifier is your PREFIX and the last qualifier is CLIST or EXEC, the first and last qualifiers are omitted. The remaining portion of the data set name, if any, and the member name is specified.

```
Execute a CLIST; for "shortnames", the last qualifier must be CLIST and
      must be omitted:
      EXEC 'fullname(member)' 'parameters'
      EXEC 'fullname(member)' 'parameters' CLIST
      EXEC shortname(member) 'parameters'
      EXEC shortname(member) 'parameters' CLIST
```

```
Execute a REXX exec; for "shortnames", the last qualifier must be EXEC
      and must be omitted; for the two examples without the "EXEC"
      keyword at the end, the first line of the exec must contain a
      comment containing the word REXX (/*REXX*/):
      EXEC 'fullname(member)' 'parameters'
      EXEC 'fullname(member)' 'parameters' EXEC
      EXEC shortname(member) 'parameters'
      EXEC shortname(member) 'parameters' EXEC
```

?? The 'parameters' may be omitted from the command line if no parameters are to be passed to the CLIST or REXX exec.
?? The CLIST keyword at the end of the command line should only be specified for CLISTs.
?? The EXEC keyword at the end of the command line should only be specified for REXX execs.
?? If neither CLIST nor EXEC is used at the end of the line, CLIST is assumed as the last qualifier for a "shortname"; however, the processor examines the first line of the file to determine whether a CLIST or REXX exec is being invoked. If the first line contains **/*REXX*/**, a REXX exec is assumed; otherwise a CLIST is assumed.

11.4.2 Implicit Execs

CLISTs and REXX execs may be executed implicitly. In this instance, the EXEC command is not used; the member name of the exec is used as if it were a command. This requires that the CLIST or REXX execs reside in libraries which are assigned to the ddnames SYSPROC or SYSEXEC.

The libraries assigned to SYSPROC may contain CLISTS and/or REXX execs. The libraries assigned to SYSEXEC may contain only REXX execs. The record format of the libraries may be fixed (F B) or variable (V B), but must match the record format of any existing libraries already assigned to the ddname. The record length may be up to 255 for fixed format or 259 for variable format, but for fixed format the record length must match any existing libraries already assigned to the ddname.

CAUTION: SYSPROC is pre-allocated to numerous system libraries. If you add any libraries to SYSPROC, be sure to include the existing system libraries when reallocating SYSPROC. The system libraries are in card-image (fixed format with record length of 80); any additional libraries should also be card-image.

CAUTION: SYSEXEC may be pre-allocated to various system libraries. If you add any libraries to SYSEXEC, be sure to include the existing system libraries when reallocating SYSEXEC. All libraries allocated to SYSEXEC should match in record format (and record length for fixed format). Also, SYSEXEC will not be searched for any REXX execs unless SYSEXEC searching is activated. The EXECUTIL command, described below, is used to turn SYSEXEC searching on or off. (At ASU, the SYSEXEC searching is on by default.)

```
Execute a CLIST or REXX exec implicitly:
      membername parameters
      %membername parameters
```

?? The parameters are optional and may be omitted.
?? The **%** is optional and may be omitted; however, use of % results in slightly improved performance. The system normally checks several libraries before the exec libraries; the % informs the system to check only the exec libraries.

11.5 EXECUTIL

EXECUTIL determines whether or not the ddname SYSEXEC is searched for implicit execs. By default, only the ddname SYSPROC is searched for implicit execs. (SYSEXEC may contain only REXX execs; SYSPROC may contain CLISTS and REXX execs.) EXECUTIL may be executed at any time to activate or deactivate searching of SYSEXEC.

```
Turn searching on:
      EXECUTIL SEARCHDD(YES)

Turn searching off:
      EXECUTIL SEARCHDD(NO)
```

11.6 FREE

FREE is used to deallocate ddnames or data sets. When attempting to reallocate a ddname, the ddname must first be freed, else the REUSE parameter is needed on the ALLOC statement. If the data set being freed is part of a concatenation list, you should free the ddname instead.

```
Free one or more ddnames:
      FREE DD(ddname)
      FREE DD(ddname1 ddname2 etc.)

Free one or more data sets:
      FREE DSN(datasetname)
      FREE DSN(datasetname1 datasetname2 etc.)

Free a print file ddname and change the destination:
      FREE DD(ddname) DEST(newdest)
```

11.7 FTP

FTP is used to transfer files from one user to another, usually between different computing systems.

```
Establish an FTP connection with another system:
      FTP ip-address
```

11.8 HELP

HELP is used to display various TSO help topics. This form of HELP does not use ISPF.

```
Display list of available help topics:
      HELP

Display a specific help topic:
      HELP topic
```

11.9 IOF

IOF is used to monitor the progress of batch jobs queued for execution, executing, and completed. IOF can also be used to cancel active jobs and to review, print, purge, or redirect output from jobs that have completed. IOF performs the same functions as the WYLBUR commands CANCEL, FETCH, LOCATE, PRINT, ROUTE, and PURGE.

```
Display batch job and print queues:
      IOF
```

(IOF can also be invoked by choosing option I from the ISPF main menu.)

11.10 ISPF

ISPF is used to start the ISPF shell, which provides a series of menus to simplify usage of TSO and provides additional functionality, including a full-screen editor. When ISPF starts, it automatically displays the primary menu; however, you can include an option to cause it to start with a different menu. Valid options are those that would be valid on the primary menu.

```
Start the ISPF shell:
      ISPF [option]
```

11.11 ISRDDN

ISRDDN is a TSO procedure that can only be invoked from an ISPF command line. It provides an ISPF screen which displays the currently allocated ddnames and data sets. Other functions (e.g., edit or browse) can be invoked from the screen.

```
Display allocated ddnames and data set names:
      ISRDDN
```

(See also LISTALC.)

11.12 LINK

LINK invokes the linkage editor to create a binary executable program from the object code data sets.

```
Link several object code files:
      LINK (objectcode1 objectcode2 etc.)

Link an object code file along with any subroutine libraries:
      LINK (objectcode) LIB(maybe-imsl-library)

Link an object code file, a subroutine library, and put the load module
      in a specific load module library:

      LINK (objectcode) LOAD(load-library-name) LIB(some-library)
```

?? One or more names of object code data sets form the first positional parameter. If the first qualifier of an object code data set is your PREFIX (typically your userid) and the last qualifier is OBJ, the first and last qualifiers may be omitted.
?? Use of IMSL or other system provided libraries require the fully qualified data set names of the libraries be included in the LIB parameter.
?? Without the LOAD parameter, LINK generates a load module name similar to the object code data set name. It's recommended that you provide a load module name, preferably one that already exists.

11.13 LISTALC

LISTALC displays the currently allocated ddnames and data sets.

```
Display allocated ddnames and data set names, except system-generated
      data set names:
      LISTALC

Display all allocated ddnames and data set names, including system-
      generated data set names:
      LISTALC SYSNAMES
```

(See also ISRDDN.)

11.14 LISTCAT

LISTCAT displays a list of cataloged files for which the first qualifier matches your PREFIX (typically your userid). The LEVEL operand is needed to display a different list. ISPF also has an option to display the similar information in a full-screen format.

```
Display catalog entries beginning with your userid:
      LISTCAT
```

```
Display catalog entries beginning with WYL.AB.CDE:
      LISTCAT LEVEL(WYL.AB.CDE)
```

11.15 LISTDS

LISTDS displays data set attribute information, such as record format, blocksize, or member names. ISPF has several more convenient methods for obtaining the same information.

```
Display data set attributes:
      LISTDS datasetname

Display member names in a partitioned data set:
      LISTDS datasetname MEMBERS
```

11.16 LOADGO

LOADGO is used to load object code into memory, link-editing it as it's loaded, and then executing it without creating a binary executable file (load module). Alternately, one of the ISPF panels provides a method for doing a similar process.

```
Load and execute several object code files with runtime parameters:
      LOADGO (objectcode1 objectcode2 etc.) 'parameters'

Load and execute object code along with any subroutine libraries:
      LOADGO (objectcode) LIB(maybe-imsl-library maybe-calcomp)
```

?? Use of IMSL and other system provided libraries require the fully qualified data set names of the libraries be included in the LIB parameter.

11.17 LOGOFF

The LOGOFF command is used to terminate the TSO session. You must exit ISPF before you can use the LOGOFF command.

```
Logoff from the current TSO session:
      LOGOFF
```

11.18 LOGON

The LOGON command is used to terminate the current TSO session and begin another one, typically with a different userid. You must exit ISPF before you can use the LOGON command.

```
Logoff from the current session and logon to a new session:
      LOGON [userid]
```

11.19 PROFILE

PROFILE is used to establish various settings for your TSO sessions. In particular, you can change your PREFIX setting, which has an initial value of your userid each time you logon.

```
Change the PREFIX to another value:
      PROFILE PREFIX(ABCDE)

Disable the PREFIX; all data set names must be fully qualified:
      PROFILE NOPREFIX

Display the current PROFILE settings:
      PROFILE LIST

Turn message identifiers on/off for diagnostic messages:
      PROFILE MSGID
      PROFILE NOMSGID
```

11.20 RECEIVE

The RECEIVE command is used to receive files sent from another user and restore them to disk. The TRANSMIT command is used to send a file to another user. RECEIVE should be specified without any options.

The RECEIVE command (with no options) prompts you for each file to be received. (You have no control over the order in which the files are received.) There are three possible responses to the prompt: **DELETE**, to delete the current file; **END**, to terminate the RECEIVE command); and **RESTORE** with optional parameters to receive the file. (RESTORE is the default and may be omitted.)

If you receive the file without providing a new name, the file is given the same name as the original but with the first qualifier replaced with your userid. To save the file with a different name, use **DSN(filename)** as one of the optional parameters.

Use the TSO Help command for further information: **`HELP RECEIVE`**. (On an ISPF command line, use **`TSO HELP RECEIVE`**.)

A log file named **userid.LOG.MISC** is written containing a summary of the files received or transmitted. The space for this file comes out of your disk quota, so you should periodically delete it.

11.21 RENAME

RENAME is used to rename data sets or members of data sets. ISPF also has several rename facilities. If a data set is protected by obsolete features, such as the NOPWREAD setting, the TSO RENAME command should be used.

```
Rename a data set:
      RENAME oldname newname

Rename a member:
      RENAME dataset(oldmembername) dataset(newmemb ername)
```

11.22 SEND

SEND is used to send a message to another user on the MVS system.

```
Send a message to several userids:
      SEND 'text to be sent' USER(userid1 userid2 etc.)
```

11.23 SUBMIT

SUBMIT is used to submit batch jobs for processing. This command is discussed in the BATCH JOB SUBMITTAL section of this write-up.

```
Submit a batch job:
      SUBMIT dsname                    /* TSO command */
      SUBMIT (dsname dsname ...)       /* TSO command */
      SUBMIT                           /* SPF/PDF editor command */
```

11.24 TRANSMIT

The TRANSMIT command is used to send files to another user. The RECEIVE command is used to receive files sent from another user and restore them to disk. (ASU's VM/CMS system has a locally developed utility, MVSSEND, which can transmit a file to the Academic MVS system. The syntax for MVSSEND is similar to the SENDFILE command. Do NOT use the SENDFILE command.)

The following commands illustrate how to transmit an entire partitioned data set, a single member of a partitioned data set, and a sequential data set. (The node name of the academic MVS system is **ACADMVS**.)

```
TRANSMIT node.userid DSN(pds.data.set)               /* entire PDS */
TRANSMIT node.userid DSN(pds.data.set(member)) SEQ   /* one member */
TRANSMIT node.userid DSN(seq.data.set) SEQ           /* sequential */
```

Use the TSO Help command for further information: **HELP TRANSMIT**. (On an ISPF command line, use **TSO HELP TRANSMIT**.)

A log file named **userid.LOG.MISC** is written containing a summary of the files received or transmitted. The space for this file comes out of your disk quota, so you should periodically delete it.

11.25 UNPRESS

UNPRESS is used to decompress WYLBUR edit format data sets. UNPRESS has one required positional parameter (the name of the data set to be decompressed) and an optional DSNAME keyword parameter (the name of the output data set). If the output data set name is omitted (recommended), the original data set is replaced with a decompressed, variable format data set.

Use the DSNAME option only if you want to retain the original data set or if you want the decompressed data set to be in fixed format. In this situation, the output data set must exist before you execute UNPRESS; i.e., you need to create an empty output data set (with a name different than the input data set) of the appropriate size and format before executing UNPRESS.

UNPRESS can be entered on a command line, but is most often used on an ISPF 3.4 data set list panel. On a data set list panel, you tab to the line containing the data set name of interest and type %UNPRESS to the left of the data set name. The data set name is assumed as the first positional parameter, i.e., you don't have to (and must not) type the input data set name.

(UNPRESS is actually an exec written in the REXX language. As such, the command is usually typed as %UNPRESS for efficiency reasons.)

```
Decompress a WYLBUR data set:
      %UNPRESS inputdsn                    /* Decompress in place */
      %UNPRESS inputdsn DSNAME(outputdsn) /* Decompress into another */
```

12. MISCELLANEOUS

12.1 Interactive Program Execution

Interaction execution varies by software product. The documentation for each software product provides the necessary information. In general, software is accessed via ISPF panels or system provided execs. Some software may require additional ALLOC statements before being used, while others may require use of the CALL statement. For example, SAS is invoked by a single command, which is an exec that internally contains many ALLOC statements and a CALL statement. It is rarely necessary to enter any other commands to begin using SAS.

For some programs, such as those written in FORTRAN, it is necessary to compile the program, (maybe) link it, and then execute it. For example, a FORTRAN program may be compiled and linked from ISPF option 4 and later executed via the CALL statement. Alternatively, you can bypass ISPF and the system execs entirely by entering the various ALLOC and CALL statements yourself, though you would need to refer to the reference manuals to determine what statements are needed.

You can create execs to automatically execute certain commands, build an edit macro, or execute one of your own programs. Typically, the exec would contain editor, ISPF, or TSO commands with some additional control statements (e.g., REXX control statements).

12.2 Technical Details

The logon procedure used by most people is ISPFPDFS, which is a member in the partitioned data set 'SYS1.PROCLIB'. ISPFPDFS defines numerous ddnames. If you want to add your own libraries to any of the pre-allocated ddnames, be sure to include the existing libraries when reallocating the ddnames.

The ddnames STEPLIB and ISPLLIB define load module libraries for binary executable programs, but only ISPLLIB may be altered during a TSO session. For example, if you put software on the system which requires a load module library, you could reallocate the ISPLLIB ddname (while ISPF is not active) to define additional runtime libraries. Thus you could subsequently invoke the software from within the ISPF environment.

Other ddnames and their uses are: SYSPROC for CLIST and REXX execs (interpreted programs); SYSEXEC for REXX execs only; ISPPLIB for panels (menus); ISPMLIB for messages; ISPSLIB for skeletons (editor templates); ISPTLIB for tables; and ISPTABL for output tables. (The ISPxxxx ddnames can be redefined but must be done while ISPF is not active.)

Using these ddnames, you can create your own panels, execs, edit macros (which are usually CLIST or REXX execs), etc. You can customize your TSO and/or ISPF sessions according to your needs.

Other methods exists for defining ddnames which provide the same function as those described above. Refer to vendor documentation for information on this and other methods of defining run-time libraries.

12.3 Local Help and Sample Code

In addition to vendor provided online help, other locally developed help files and code samples are available for use with the MVS system.

The locally developed help files are available via a web browser (such as Netscape or Internet Explorer). The URL for the help files is **http://www.asu.edu/it/fyi/mvs/**.

Code samples for common disk/tape utility program are available during a TSO ISPF editing session. Follow the steps outlined below to access the code samples.

?? Begin editing the file into which the code sample is to be inserted.
?? Type the editing command "MODEL CLASS JCL" on the editor command line and press Enter. This identifies which set of sample code to use.
?? Type the editing command "MODEL" on the editor command line, and type an "A" or "B" in a line number field (to indicate where the sample code is to be inserted) and press Enter.

?? Select from the displayed menu whether you want Disk or Tape samples, and press Enter.
?? Select from the displayed menu which sample code to use, and press Enter. (There are more code samples than can be displayed on a single menu. Press PF7 or PF8 to see other menu pages.)

The sample code includes comments to indicate what JCL modifications should be made before the code is submitted for execution.