Unit 2: Basic Concepts

Objectives

Communicating With MVS
Using TSO/E Commands
- **Using Command Operands**
- **Abbreviating Commands and Keyword Operands**
- **Separating Words in a Command**
- **Continuing a command on Another Line**
- **Including Comments**
- **Using Subcommands**
- **Issuing TSO/E Commands**
- **Getting Help For Commands**
- **Listing All TSO/E Commands**

Using Data Sets
- **TSO/E Data Set Naming Rules and Conventions**
- **Entering Data Set Names**

Figure: 2-1. Objectives

Communicating With MVS

TSO/E allows you to communicate with the MVS operating system to do work. Ways to communicate with MVS are through:

- **TSO/E Commands: Commands control your access to the system, determine your terminal characteristics while you are on the system, allow communication between TSO/E users, and manage data sets.**

- **ISPF/PDF Panels: Interactive System Productivity Facility (ISPF) provides the underlying dialog management service for the ISPF/Program Development Facility (ISPF/PDF). ISPF/PDF is a dialog that allows a TSO/E user to issue TSO/E commands directly or indirectly from panels.**

- **Programs: Programs contain instructions that perform tasks.**

- **JCL : Job Control Language defines work (jobs) for an operating system. A TSO/E terminal session is considered a job.**

Figure: 2-2. Communicating With MVS

Using TSO/E Commands

A TSO/E command consists of a command name that is generally followed by one or more operands. A Command name is typically familiar English word that describes the function of the command; For instance, the RENAME command changes the name of the dataset.

Operands provide the command with specific information. For example, The RENAME command has two operands; one that identifies the data set to be renamed and one that specifies the new name.

```
RENAME  old.data   new.data
          / \          / \
Command  operand1  operand2
```

Figure: 2-3. Using TSO/E Commands

Notes :

Two ways to enter commands are:

- Enter the command name and let the system prompt you for required information, or
- Enter the command name and the operands at the same time

When you enter a command, however,you can type the letters as uppercase or lowercase.

Using Command Operands

There are two types of operands used with commands :

- **Positional**
- **Keyword**

Figure: 2-4. Using Command Operands

Positional Operands

Positional operands are required and must immediately follow the command name in a certain order. If you enter positional operands incorrectly, you get an error message. For example, the RENAME command requires that the first operand be the old data set name and the second operand be the new data set name .

To enter a positional operand that is a list of several names or values, enclose the list within parentheses and separate the operands by a comma or space.

Example
To list the data set attributes of more than one data set, enter the LISTDS command with the positional operands, which are the data set names, enclosed in parentheses.

LISTDS (new. data test. Data)

Figure: 2-5. Positional Operands

Keyword Operands

Keyword operands are specific names or symbols that have a particular meaning to the system. You can include keywords in any order following the positional operands. In some cases you specify values with a keyword by entering the value within parentheses following the keyword. Some keywords are not followed by a value within parentheses.

If you enter mutually exclusive keywords, the last keyword entered overrides the previous ones.

Figure: 2-6. Keyword Operands

Notes:

Example

To transmit data set NEW.DATA to USER5 at NODEID, you use a positional operand to specify the destination node and user ID and the keyword operand, DATASET, to specify the data set name. The keyword NOLOG indicates that you do not want a record of the transaction.

```
TRANSMIT nodeid.user5 DATASET (new. data) NOLOG
                      / \             / \           / \
 Positional keyword keyword operand operand operand
```

Abbreviating Commands and Keyword Operands

Abbreviating Commands

Nearly all TSO/E commands have abbreviations that you can use in place of the full command name. These abbreviations save your entry time at the terminal. If a command can be abbreviated, it has only one accepted IBM abbreviation. For example, ALLOC is the abbreviation for ALLOCATE and E is the abbreviation for EDIT.

Abbreviating Keywords

Like commands,nearly all keywords have abbreviations.You may enter keywords spelled exactly as they are shown, or you may use an acceptable abbreviation. An acceptable abbreviation is as much of the keyword as is necessary to distinguish it from the other keywords of the command or subcommand.

Notes:

- If you supply an abbreviation which is not specific enough, the system will prompt you. An exception is the TSO/E HELP command that requires you to enter an operand unambiguously for proper results.

- For readability and clarity in this book,most references to commands and explames of their use appear in the long form.

Separating Words in a Command

When you type a command, separate the command name from the first operand by on or more blanks. Separate operands from each other by one or more blanks or a comma.

For example, you can type the LISTBC command like this:

LISTBC NOMAIL NONOTICES

Or

LISTBC NOMAIL,NO NOTICES

A list of items may be enclosed in parentheses and separated by blanks or commas.

For example,

LISTBC(mydsa mydsb,mydsc)

Continuing a Command on Another Line

When you type a command at the READY message or in ISPF/PDF option 6, you can continue to type beyond the end of the line and the command automatically wraps around to the next line. However, when you type a command in a CLIST that does not fit on one line,use a plus or minus sign(preceded by a space) as the last character of the first line to continue the command onto the next line. A plus sign removes leading spaces from the continuation line. A minus sign keeps leading spaces with the continuation line.

Example:

To display a command to the system as a single line without leading spaces, type :
ALLOCATE DATASET(outputds.text) +
LIKE(old.text)
The system sees:
ALLOCATE DATASET(outputds.text) LIKE(old.text)

Using Subcommands

Some TSO/E commands such as EDIT,OUTPUT, and TEST have subcommands. Subcommands are similar to commands, they can have positional and keyword operands.

When you issue a command that has subcommands, the system responds by displaying a special mode message. You can then enter a subcommand to specify the particular operation that you want to be performed. When you want to stop entering subcommands, enter the END subcommand.

Example:
When you issue an EDIT command, instead of READY, the EDIT mode message appears, which indicates that you can enter subcommands of EDIT.

Notes:

- **READY and EDIT are mode messages.**
- **EDIT is also a command.EMODE is an operand of the EDIT command specifying that EDIT should start in EDIT mode, not INPUT mode.**
- **List, save, and end are EDIT subcommands.**

Issuing TSO/E Commands

You issue TSO/E commands or subcommands by:

- **Entering them at your terminal**
  - **After the READY mode message**
  - **On the COMMAND/OPTION line of an ISPF/PDF panel and are preceeded by the characters ‘tso’.**
  - **By using the ISPF/PDF COMMAND option(option 6)**
  - **Submitting them in a job using JCL statements.**

Entering More than one Command

You can sometimes save time by entering two or more commands separated by field mark symbols (;).

For example, if you enter the following ALLOCATE, RENAME, and DELETE commands without waiting for the intervening mode messages,

Your display is:
READY
ALLOCATE DATASET(tempdata) LIKE(other.data);
RENAME temp.data new.data;
DELETE temp.data
READY
READY
READY

Figure: 2-12. Entering More than one Command

Notes:

- Be careful when entering commands without waiting for the intervening mode messages. If you make a mistake in one of the commands, the system sends you one or more messages, and then cancels the remaining commands you entered. After you correct the error, you have to reenter the other commands. Therefore, unless you are sure your input is correct, wait for a READY message before entering a new command.
- Some terminals lock the keyboard after you enter a command, so that when you press the keys, you are not communicating with the system. You cannot enter commands until you get a READY message. Terminals that do not normally lock the keyboard might occasionally do so, for example, when all buffers allocated to the terminal are full.

Getting Help for Commands

An informational message tells you about the status of the system or of your terminal session. You need not respond to an informational message. Informational messages include a message identifier (message ID) and are documented in TSO/E Messages.

Like prompting messages, informational messages may have two or more levels. If an informational message ends with a plus sign (+), you can request an additional message by entering a question mark (?) in the leftmost space under the message as described previously in "Prompting Messages".

Figure: 2-13. Getting Help for Commands

Getting Help for Commands (Contd.)

To display information about any TSO/E command and about some of the command's messages, use the HELP command.

By using the HELP command with its operands, you can request that the system display :

- **A list of all IBM-supported TSO/E commands in the system, along with an explanation of each**

- **Information about a particular command, such as:**
  – The function and operation of a command
  – The operands used with a command.

Figure: 2-14. Getting Help for Commands (Contd.)

Notes:

- The EDIT, TEST, and OUTPUT commands have a HELP subcommand. This subcommand acts similarly to and uses similar syntax as the TSO/E HELP command. When using the HELP subcommand, follow the instructions for the HELP command, but substitute the subcommand names.

Listing All TSO/E Commands

To display a list of all the TSO/E commands in the system along with a description of each, enter the HELP command with no operands.

Example
To display all TSO/E commands, enter: HELP
You then see information similar to the following:

LANGUAGE PROCESSING COMMANDS:
ASM INVOKE ASSEMBLER PROMPTER AND ASSEMBLER F COMPILER.
CALC INVOKE ITF: PL/1 PROCESSOR FOR DESK CALCULATOR MODE.
COBOL INVOKES COBOL PROMPTER AND ANS COBOL COMPILER.
FORT INVOKE FORTRAN PROMPTER AND FORTRAN IV G1 COMPILER . . ..

An installation may also place its own help information about installation-written commands on the system.

Figure: 2-15. Listing All TSO/E Commands

Notes :

Requesting Information about a Command
You can request to see information about a particular command by specifying the HELP command followed by the command name.
Example
To see help information about the RENAME command, enter :
HELP RENAME

Using Data Sets

A data set is a unit of information that can be stored and retrieved. Some types of data you might put into a data set are :

- **A program's source code**
- **Job control language statements**
- **Input to a program**
- **Output from a program**
- **Text of a report.**

A data set is organized in one of several arrangements and is described by control information that the system can access. The two types of data sets most often used with TSO/E are :

- **Sequential data set - A single unit with data arranged in a sequence, from top to bottom or beginning to end.**
- **Partitioned data set (PDS) - Subdivided unit that is divided into separately named, independent partitions called members, each of which can contain data. A partitioned data set has a directory that contains information about each member.**

Figure: 2-16. Using Data Sets

Data Set Naming Conventions

In addition to the rules, use of certain naming conventions that will make TSO/E easier for you to use. These conventions are an offshoot of the rules. Thus a data set name that follows the naming rules might not follow the naming conventions.

The conventions are :

- **Data set names consist of three qualifiers.**
- **The first qualifier of each data set name is your prefix as specified in your user profile. Sometimes your prefix is your USERID.**
- **The second qualifier of each data set name is your choice; it should be a meaningful name to you.**
- **The third qualifier is a descriptive qualifier implying certain characteristics of the data.**

Figure: 2-17 Data Set Naming Conventions

Note:

A data set name that consists of a prefix, a user-supplied name, and a descriptive qualifier is a “fully-qualified” data set name. A fully-qualified data set name looks like: When you refer to partitioned data sets, enclose the member name in parentheses immediately following the descriptive qualifier. You do not have to use the conventional descriptive qualifiers when naming a TSO/E data set. However, when a data set name adheres to the conventions, you can refer to the data set by an abbreviated version of the name, and the system supplies the rest of the name.

Entering Data Set Names

The data set naming conventions simplify the use of data set names. If the data set name confirms to the conventions, you need to specify only the user-supplied name field when you refer to the data set in commands.

When you specify only the user-supplied name, the system adds the prefix specified in your user profile, sometimes your USERID and, whenever possible, a descriptive qualifier. The system attempts to derive the descriptive qualifier from available information.

The system adds your prefix as the first qualifier and adds the data set type data because that is the data set type assigned to PREFIX.TEST5.DATA. In some cases, however, the system prompts you for a descriptive qualifier. If you do not specify enough information, the system issues a message at your terminal requesting the required information.

Figure: 2-18. Entering Data Set Names

Notes: The system will not append qualifiers to any name enclosed in single quotation marks.

Some of the commands used are :

- **Call**
- **Allocate**
- **Delete**
- **Edit**
- **Rename**
- **Listcat**
- **Time**
- **Logon**

Figure: 2-3. Using TSO/E Commands

Call

FUNCTION: -

The CALL command loads and executes programs in executable form.

SYNTAX -

CALL ‘dsname (member)' 'parm'
CAPS/ASIS NOENVB/PASSENVB

Figure: 2-4 Call

Notes:

CAPS: The CAPS operand indicates that the parameter string 'PARM' is to be translated to uppercase before the called program executes. This is the default.

ASIS: The ASIS operand indicates that the parameter string 'PARM' should not be translated to uppercase before the called program executes.

NOENVB: The NOENVB operand indicates that the REXX environment block (ENVBLOCK) address is not to be passed to the called program in register 0. This is the default.

PASSENVB: The PASSENVB operand indicates that the REXX environment block (ENVBLOCK) address is to be passed to the called program in register 0.

‘dsname (member)' : The name of a partitioned data set and the member which is to be executed
'Parm’ - Specifies a parameter string (up to 100 characters) that is passed to the program to be executed.

Allocate

FUNCTION: -

The allocate command performs the following:

Dynamically defines and allocates a data set with or without an attribute list of DCB parameters.
Allocates a new dataset with attributes copied from an existing data set.
Concatenates a list of data sets.
Allocates an HFS file.

SYNTAX -
ALLOCATE DATASET ('DSNAME'/'LIST OF DSNAMES'/*)
OR
DUMMY

Figure: 2-5. Allocate

Notes:

OPERANDS –

DATASET (‘DSNAME (S)'/*) OR DSNAME (‘DSNAME (S)'/*) - Dsname is name of data set to be allocated. * Indicates terminal to be allocated.

DUMMY - Dummy data set to be allocated

Delete

FUNCTION: -

The DELETE command is used to delete either VSAM objects or NONVSAM Data sets from a VSAM or ICF catalog and to free space occupied by the objects or data sets. Also the free space of VSAM data components can be overwritten with zeros. The DELETE command also deletes GDG bases and VSAM USERCATALOGS.

DELETE ('entry name/password'...)
CATALOG ('catname/password')
FILE ('dname')
PURGE ¦ NOPURGE
ERASE ¦ NOERASE

Figure: 2-6. Delete

Notes:

Operands –
'entry name/password' - Specifies the name of entry to be deleted.

CATALOG ('catname/password') -Specifies the name of catalog containing the entries to be deleted.

'Catname' - Name of catalog containing the entries to be deleted.

'Password' - Master level password of the catalog.

FILE ('dname') - Specifies the location of the object to be deleted.

'dname' - Name of the DD statement that identifies either the volume containing the object to be deleted or the data set to be deleted.

Edit

FUNCTION: -

The edit command is used to create or modify sequential data sets or Members of partitioned data sets.

SYNTAX -

EDIT ‘DSNAME’ NEW/OLD SCAN/NOSCAN

Figure: 2-7. Edit

Notes:

OPERANDS –

'DSNAME' - Name of the data set to be created or edited.

NEW - Data set named did not exist before command was issued.

OLD - Data set already existed when the edit command was issued.

SCAN - For data set type or each line is to be checked for syntax.

NOSCAN - No syntax checking is to be performed.

Listcat

FUNCTION: -

The LISTCAT command lists entries from either the master catalog or a user catalog.

Syntax -

LISTCAT CATALOG ('catname/password')
LIBRARY ('lib name')
FILE ('dname')
OUTFILE ('dname')

Figure: 2.8. Listcat

Notes:

Operands –

CATALOG ('catname/password') - Specifies the name of the catalog containing the entries to be listed.

'Catname' - Name of the catalog containing the entries to be listed.
'Password' - Password of the catalog containing the entries to be listed.

LIBRARY ('library name') - Specifies the name of the library that is used as a filter to restrict the volume entries to be listed.
'library name' - Name of the library that is used as a filter to restrict the volume entries to be listed.

FILE ('dname') - Identifies the volumes that contain the catalog entries to be listed.
'dname' - Name of the DD statement that identifies the volumes containing catalog entries to be listed.

OUTFILE ('dname') - Identifies the alternate output data set.

Rename

FUNCTION: -

The rename command is used to rename a data set or a partitioned data set member or to create an alias for a partitioned data set member.

SYNTAX -

RENAME 'DSNAME1' 'DSNAME2' ALIAS
REQUIRED - 'DSNAME1' AND 'DSNAME2' DEFAULTS - NONE

Figure: 2-9. Rename

Notes:

OPERANDS –

'DSNAME1'- The current data set name.

'DSNAME2'- The new data set name to be assigned.

ALIAS - Member name specified by 'DSNAME2' is to be an alias rather than a replacement.

Time

FUNCTION: -

The time command informs the terminal user of the local time of day, the date, cumulative CPU time, service units, and the total time the the user has been logged on the terminal during a session.

SYNTAX –

TIME

Figure: 2.10. Time

Notes:

OPERAND – NONE

REQUIRED – NONE

DEFAULTS – NONE

NOTE - ANYTHING IN THE OPERAND FIELD WILL BE IGNORED.

Logon

FUNCTION: -

The logon command initiates a terminal job.

SYNTAX -

LOGON 'ID' ACCT ('ACCOUNT') PROC ('PROCEDURE') SIZE ('INTEGER')

Or

LOGON ‘ID’ RECONNECT

Figure: 2-11. Logon

Notes:

OPERANDS -
'ID' - is a user's identification to the system. The ‘id’ consists of 'userid/current-Password/new-password’. If the user does not have passwords then '/current-password' is optional and '/new-password' is ignored. '/New-password' is valid only for RACF users. It will become the '/current-password' the next time the user logs on. It will be ignored for non-racf users.

ACCT ('ACCOUNT') - 'Account' is the information required by the installation accounting routines.

PROC ('PROCEDURE') – ‘Procedure' is the name of a catalogued procedure, which identifies the system resources required for a terminal job.

SIZE ('INTEGER') - 'Integer' is the upper bound on the size of a variable conditional get main which can be satisfied.

Unit Summary

“Basic Concepts” explains things you need to know to use TSO/E, such as :

- **Commands - TSO/E command syntax and how to issue commands**
- **Data sets - types of data sets and how to name them.**

Figure: 2.19 Unit Summary