

**Chapter
2**

**TSO
COMMANDS**

*Get on the
Fast Track!*



TM

**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- Executing TSO commands in READY mode or ISPF.
- The format of a TSO command - syntax and usage.
- Allocating a dataset based upon another dataset.
- Commonly used TSO commands.
- TSO commands - summary list.

1 TSO/E Commands

Commands and derivative forms of commands are used for communicating with and managing through TSO/E.

TSO/E line mode commands consist of the command name, with optional operands.

- The command is typically the word which describes the purpose and functionality of the command.
- Operands provide specific information.

Commands can have two kinds of operands:

positional	keyword
------------	---------

With ISPF screens, commands are entered in a different form by selecting options and entering information in fields. ISPF screens also have their own commands, such as the PF key commands and keys.

1.1 Positional Commands

The RENAME command changes the name of a dataset. The operands are the old dataset name and the new dataset name.

When a command has positional operands, they will need to be specified with the command. Positional operands must follow the command in a certain order.

With the RENAME command, the old dataset name must be positioned first and the new dataset name must be positioned second.

Example:

```
RENAME FILE1.DATA FILE2.DATA
```

FILE1.DATA	operand1
FILE2.DATA	operand2

1.2 Keyword Commands

Keyword operands are names or symbols that have a specific meaning to the system. They can be in any order, but cannot appear before positional operands.

The ALLOC command can utilize the DATASET keyword. The word DATASET is a keyword that is followed by the name of the dataset enclosed within parentheses.

Example:

```
ALLOC DATASET(ABC.DATA) ...
```

2 Syntax and Usage Rules

The syntax rules for TSO commands are:

- The command name must always be first.
- Positional operands must follow next after the command.
 - If there is more than one positional operand, the operands must be in a particular order.
- Keyword operands, when utilized, can be in any order after positional operands.
 - If more than one item is within the parentheses following a keyword, the items must be separated by a comma or a space.

2.1 Entering Commands

TSO/E commands can be entered:

- After the READY message.
- From the ISPF COMMAND option: option 6.
- On any ISPF/PDF panel by typing the TSO/E command on the COMMAND or OPTION line preceded by TSO.

3 Dataset - Types

TSO/E utilizes and manages both sequential datasets and partitioned datasets.

The content of a sequential dataset resides in a single file in sequential order. These files can have one or more names separated by periods such as `BILLS`, or `CURRENT.BILLS`, or `MY.CURRENT.BILLS`.

This listing is an example of a dataset.

Example:

00001	1/04	Groceries	\$42.21	Paid
00002	1/05	Gasoline	\$18.00	Paid
00003	1/13	Towing charge	\$25.00	Unpaid
00004	1/14	Snow tires	\$70.95	Unpaid
00005	1/16	Groceries	\$67.14	Paid
00006	1/16	Gas and electric	\$112.52	Unpaid
00007	1/17	Telephone	\$80.05	Unpaid

When the file for bills becomes large and difficult to find information, it can be organized into subcategories. The bills in this file can be divided into 'paid' and 'unpaid'. Two new folders can be created with the new category names. Both folders will remain part of the general category 'bills'.

TSO/E manages subdivided files with partitioned datasets, which are referred to as a PDS or a library. A PDS can contain one or more members. As with a sequential dataset, the content is arranged in sequence.

Partitioned datasets generally have three names separated by periods and a member name in parentheses.

Example:

```
MY . CURRENT . BILLS (PAID)
MY . CURRENT . BILLS (UNPAID)
```

4 Dataset Naming - Rules

An IBM z/OS mainframe operating system imposes rules for naming datasets.

The following rules apply to naming TSO/E datasets:

- A dataset name consists of one or more parts connected by periods.
 - Each part is called a qualifier.
 - Each qualifier must begin with an alphabetic character (A to Z) or the special character @, #, or \$.
 - The remaining characters in each qualifier can be alphabetic, special, or numeric (0 to 9) characters.
 - Each qualifier must be 1 to 8 characters in length.
- The maximum length of a complete dataset name before specifying a member name is 44 characters, including the periods.

4.1 Dataset Naming Conventions

In addition to the rules, there are naming conventions for defining a dataset to ISPF/PDF.

Conventions provide standards that make it easier to use TSO/E whether in ISPF/PDF or in line mode TSO/E.

Both sequential and partitioned dataset names should consist of three qualifiers.

First qualifier	PROJECT	Is the prefix as specified by the PROFILE command? Sometimes a prefix will be a TSO/E user ID.
Second qualifier	GROUP	Can be any meaningful name.
Third qualifier	TYPE	Description of the type of information in the dataset such as data, text, or the name of a programming language.

Example:

An ISPF/PDF dataset with a member:

```
PROJECT  ==> myid
GROUP    ==> claims
TYPE     ==> data
MEMBER   ==> feb
```

5 Dataset Allocation - Based on an Existing Dataset

The syntax of the ALLOCATE command provides for specifying a large number of dataset attributes. Accordingly, allocating a dataset based upon an existing dataset is convenient and useful.

A dataset can be created and named PREFIX.NEWTEST.DATA with the same attributes as the one named PREFIX.TEST.DATA.

Example:

After the READY message enter:

```
ALLOCATE DATASET(newtest.data) LIKE(test.data)
```

The word prefix has not been included as the first qualifier. TSO/E assumes that the prefix is the first qualifier unless the dataset is placed within single quotes.

Example:

After the READY message enter:

```
ALLOCATE DATASET('prefix.newtest.data') LIKE('prefix.test.data')
```

This syntax is also correct and will create the identical dataset.

6 Commands

6.1 LISTCAT Command

It is a common practice to examine datasets. The datasets will have the user's prefix as the first qualifier.

The LISTCAT command is used to display a list of datasets:

```
LISTCAT
```

The LISTCAT command provides functionality comparable to the DIR command used in Microsoft Windows on Intel-based microcomputers.

6.2 DELETE Command

The DELETE command is used for deleting one or more dataset entries or one or more members of a partitioned dataset.

The catalog entry for a partitioned dataset is removed only when the entire partitioned dataset is deleted.

The system deletes a member of a partitioned dataset by removing the member name from the directory of the partitioned dataset.

The command should be used with caution.

The syntax of the command is:

```
DELETE dsname
```

Depending on the installation, there may be a prompt as part of performing the delete operation.

Example:

```
DELETE MYDATA.DATA
```

or

```
DELETE 'MYID.MYDATA.DATA'
```

6.3 LISTDS Command

The LISTDS command is used to display the attributes of specified datasets at a terminal.

Depending upon whether the dataset is VSAM or non-VSAM, the LISTDS command will work differently. Entering the LISTDS command without operands will indicate whether a dataset is VSAM or non-VSAM.

A VSAM dataset causes the LISTDS command to display only the dataset organization, DSORG, which is VSAM.

The LISTCAT command is used to obtain more information about a VSAM dataset.

The following information can be obtained about non-VSAM datasets:

- The volume serial number of the DASD volume on which the dataset resides.
- The LRECL: logical record length.
- The BLKSIZE block size.
- The RECFM record format.
- The DSORG dataset organization -
 - PS for sequential
 - PO for partitioned
 - IS for indexed sequential
 - DA for direct access
 - VSAM for VSAM data entries
 - ** for unspecified

6.4 SEND Command

The SEND command or the SEND subcommand of EDIT can be used for sending a message to anyone at the following destinations:

- One or more users.
- An operator specified by route code.
- An operator console specified by name.

SEND can be used to send a message from one user to another user.

By default, when the SEND command is issued with the NOW operand, the message is displayed on the recipient's screen if he or she is logged on and receiving messages. If the receiver is not logged on or is not receiving messages, the message is deleted and a message is generated stating why the message has not been displayed.

The syntax of the command is:

```
SEND 'message' USER(userid)
```

Example:

```
SEND 'HELLO WORLD' USER(B300)
```

6.5 TSO E/Commands - Summary

Command	Function
ALLOCATE	Dynamically allocates datasets.
ALTLIB	Defines alternative application-level libraries of REXX EXECs or CLISTs.
ATTRIB	Builds a list of attributes for non-VSAM datasets.
CALL	Loads and executes a program.
CANCEL	Ends the processing of batch jobs submitted at a terminal.
DELETE	Deletes dataset entries or members of a partitioned dataset.
EDIT	Creates, modifies, stores, submits, retrieves, and deletes datasets. There are also EDIT subcommands.
END	Ends a CLIST.
EXEC	Executes a CLIST or REXX exec.
EXECUTIL	Changes various characteristics that control how REXX execs run in the TSO/E address space only.
FREE	Releases previously allocated datasets, changes the output of a SYSOUT dataset, deletes attribute lists, or changes dataset disposition.
HELP	Provides information about the function, syntax, and operands of commands and subcommands and information about certain messages.
LINK	Invokes the linkage editor service program.
LISTALC	Lists datasets that are currently allocated to the TSO/E session.
LISTBC	Displays messages of general interest.
LISTCAT	Lists entries from a catalog by name or entry type.
LISTDS	Displays attributes of datasets.
LOADGO	Loads a compiled or assembled program into real storage and begins execution.
LOGOFF	Ends the terminal session.
LOGON	Starts a terminal session.
OUTDES	Creates or reuses dynamic output descriptors.
OUTPUT	Directs output from a job at a terminal or to a specific dataset; deletes the output, changes output class, routes output to a remote workstation, or releases the output for a job for printing by the subsystem.
PRINTDS	Formats and prints datasets on any printer defined to JES.
PROFILE	Changes or lists a user profile.
PROTECT	Prevents unauthorized access to non-VSAM datasets.
RECEIVE	Retrieves transmitted files and restores them to their original format.
RENAME	Changes the name of a non-VSAM cataloged dataset, changes the member name of a partitioned dataset, or creates an alias for a partitioned dataset member.
RUN	Compiles, loads, and executes the source statements in a dataset.
SEND	Sends a message to another terminal user or to the system operator.
SMCOPY	Copies all or part of a stream or dataset to another stream or dataset.
SMFIND	Locates a string of characters in a stream.
SMPUT	Places a string of characters in a stream.
STATUS	Displays the status of a job.
SUBMIT	Submits one or more batch jobs for processing.

Command	Function
TERMINAL	Lists or changes operating characteristics of the terminal.
TEST	Tests a program or command processor written in Assembler language.
TIME	Displays CPU and session time, total service units used, local time of day and date.
TRANSMIT	Sends information, such as a message or a copy of information in a dataset, to another user in the network.
TSOEXEC	Invokes an authorized command from an unauthorized environment.
TSOLIB	Dynamically links to different versions of load module libraries from within a user's TSO/E session.
WHEN	Tests return codes from programs invoked from an immediately preceding CALL or LOADGO command, and to take prescribed action if the return code meets a specified condition.