

# Time Sharing Option Extension (TSO/E)

<b>INTRODUCTION TO TSO/E</b> .....	<b>2</b>
WHAT IS TSO/E?.....	2
<i>Features of TSO/E</i> .....	2
FUNCTIONS OFFERED BY TSO/E.....	2
LOGON COMMAND.....	2
STEPS FOR LOGGING-ON.....	2
<i>What after LOGON?</i> .....	2
ISPF/PDF.....	3
<i>Line mode TSO/E</i> .....	3
<i>Steps For Logging-off</i> .....	3
COMMAND SYNTAX.....	3
COMMAND SYNTAX.....	3
DATASET MANAGEMENT.....	4
ALLOCATE COMMAND.....	4
<i>ALLOCATE Command - OPERANDS</i> .....	4
<i>ALLOCATE Command - OPERANDS</i> .....	4
FREE COMMAND.....	6
<i>Release of Specific Datasets</i> .....	6
LISTALC COMMAND.....	7
<i>Operands of LISTALC</i> .....	7
<i>LISTALC History Operand</i> .....	7
LISTCAT OPERANDS.....	8
LISTDS COMMAND.....	9
<i>LISTDS MEMBERS operand</i> .....	9
RENAME COMMAND.....	10
SMCOPY COMMAND.....	10
PRINTDS COMMAND.....	11
DELETE COMMAND.....	11
COMMUNICATE WITH OTHER USERS IN TSO/E.....	12
TRANSMIT.....	12
RECEIVE.....	12
RUNNING/EXECUTING PROGRAMS.....	13
<i>Foreground mode --</i> .....	13
<i>BACKGROUND</i> .....	13
<b>TSO COMMAND SUMMARY</b> .....	<b>14</b>

# INTRODUCTION TO TSO/E

## ***What is TSO/E?***

- Time Sharing Option Extensions.
- An option of the MVS Operating System that allows users to interactively share computer resources.

## **Features of TSO/E**

TSO/E can be used to

- Communicate with other TSO/E users.
- Develop and maintain programs.
- Process Data.
- Access the MVS Operating System.

## ***Functions Offered by TSO/E***

- Logon/Logoff Procedures.
- Dataset Management.
- Communication.
- Running/Executing Programs.

## ***Logon Command***

Starts a TSO/E Session and identifies the user to TSO/E.

User Id:

- Maximum Seven Characters.
- Can contain numeric(0-9) and Alphabetic(A-Z) Characters.
- First letter must be Alphabetic.

Password:

- Group of one to Eight Characters.

## ***Steps For Logging-on***

- Enter the LOGON Command.
- System prompts you to give the userid and password.
- On successful entry, System prompts - READY.
- The prompt READY means TSO/E is ready for commands. This is the line mode of TSO/E.

## **What after LOGON?**

- Can use line mode TSO/E commands.

- Use the ISPF/PDF facility.

## ***ISPF/PDF***

- Interactive System Productivity Facility / Program Development Facility.
- It is a screen/panel based facility to manage information on the MVS System.

## **Line mode TSO/E**

- It is a command language that interacts directly with the MVS Operating System.

## **Steps For Logging-off**

- Logging off ends the TSO/E session.
  - Enter the LOGOFF Command on the READY prompt.
  - Press the ENTER key.
  - System gives the following message :

<userid> LOGGED OFF TSO AT 11:15:45 ON JUNE 5, 1997

## **Command Syntax**

TSO/E line commands consists of :

- Command Name
- Operands (Positional or Keyword)

**Positional operands:** must follow the command in a fixed order.

**Keyword operands:** are identified by symbols that have specific meaning.

Where can Commands be given?

- READY Prompt.
- In ISPF/PDF option 6 panel.
- On any ISPF/PDF panels command line.

## **Command Syntax**

Examples : -

1. RENAME OLD.DATA NEW.DATA  
(Command) (Operand 1) (Operand 2)
2. TRANSMIT nodeid.users DATASET(new.data) NOLOG  
(Command) (Pos Operand) (Keyword opd 1) (Keyword opd 2)

All Positional Operands have to precede Keyword Operands. Keyword Operands can be in any order.

## ***DATASET MANAGEMENT***

- Datasets contain information that can be stored and retrieved.
- TSO/E manages simple files - Sequential datasets
- TSO/E manages subdivided files, Partitioned datasets which contain one or more members.

Following Commands and Functions are used to manage datasets.

ALLOCATE	- Create new datasets
FREE	- Release datasets
LISTCAT	- List datasets
LISTDS	- List information about datasets
RENAME	- Change name of datasets or members
SMCOPY	- Copy datasets or members
PRINTDS	- Print datasets
DELETE	- Delete datasets or members

### ***ALLOCATE Command***

Allocation is the process of requesting access to a dataset.

It allows the user to :

- Open an existing dataset  
If you Allocate a dataset that exists, the system allows you to open and use the dataset.
- Define a new dataset  
If you Allocate a dataset that is new, the system creates space for it and allows you to open the dataset.

### ***ALLOCATE Command - OPERANDS***

Allocate command has many operands. Following are the operands used to create a new dataset.

DATASET name  
NEW  
DIR (optional)  
space in TRACKS or CYLINDERS  
DSORG, RECFM, LRECL, BLKSIZE  
CATALOG

Example:-

```
ALLOCATE DATASET (new.data) NEW DIR(4)
TRACKS SPACE(10,5)
DSORG(PO) RECFM(FB) LRECL(80)
BLKSIZE(3200) CATALOG
```

### ***ALLOCATE Command - OPERANDS***

Operands for accessing an existing data set.

DATASET  
FILE  
OLD/SHR/MOD (Type of access to the dataset)  
KEEP/DELETE (Disposition of the dataset)

File Operand : specifies the ddname that connects the dataset to the program or utility.  
OLD/SHR/MOD Operands: defines the type of access for the dataset.  
Disposition Operands: defines what happens to the dataset when it is unallocated.

Examples:-

```
ALLOCATE DATASET(input.data) FILE(myfile)
The default access mode is OLD.
```

```
ALLOCATE DATASET(input.data) FILE(myfile)
SHR DELETE
On doing a FREE for 'myfile' the dataset is deleted.
```

```
ALLOCATE FILE(indata) DATASET(a.data,b.data)
This allows more than one dataset as input to the program.
```

```
ALLOCATE FILE(indata) DATASET(b.data,a.data)
SHR REUSE
This command unallocates the DDNAME 'indata' and reallocates the datasets in reverse order.
```

Other ways of allocating datasets :

- Using ISPF/PDF option 3.2
- Using Job Control Language(JCL) statement

Example:-

```
//MYFILE DD DSN=NEW.DATA,DISP=(NEW,CATLG),
//        SPACE=(TRACKS,(10,5,4)),UNIT=TEMPDA,
//        DCB=(DSORG=PO,LRECL=80,BLKSIZE=3200,
//        RECFM=FB)
```

Using IBM Supplied Utility program - IEFBR14

## ***FREE Command***

- Releases the datasets that were previously allocated in a TSO/E Session.
- Releases the ddnames that link the datasets to programs.

Example : -

```
FREE ALL
```

Releases all datasets and ddnames not currently in use

## **Release of Specific Datasets**

It is possible to release previous allocations of

- A specific dataset.
- Group of datasets.
- ddname associated with specific datasets.

Examples : Freeing datasets

```
FREE DATASET(parts.data)
```

```
FREE DATASET(parts.data,report.file(mem1))
```

Releases the specified dataset(s).

Examples : Freeing ddnames

```
FREE FILE(myfile)
```

```
FREE DDNAME(sysut1,sysprint)
```

Releases the dataset(s) associated with ddnames - 'sysut1' 'sysprint'.

More Examples: -

```
FREE FILE(sysprint) SYSOUT DEST(DELHIPT1)
```

Releases a SYSOUT dataset and sends it to another location for output processing.

```
FREE FILE(sysut1) HOLD
```

Releases a dataset and place it in held queue for viewing or printing later.

```
FREE FILE(myfile) DELETE
```

On release of dataset(s), disposition options KEEP/DELETE/CATALOG/UNCATALOG are used to control what is done with the dataset(s). In the last example all the dataset(s) associated with the ddname 'myfile' are deleted.

## **LISTALC Command**

Used to obtain a list of the datasets allocated during the current TSO/E Session.

### **Operands of LISTALC**

STATUS  
HISTORY

LISTALC STATUS : For each dataset allocated to TSO/E session gives the ddname and disposition of the dataset(s).

Example: -

LISTALC STATUS  
Gives a list as follows:

```
SYS1.TSO.CLIST
  SYSPROC      KEEP
ISP.ISPF.CLISTS
              KEEP
PREFIX.ISPF.PROFILE
  ISPPROF     KEEP,KEEP
```

### **LISTALC History Operand**

Used to obtain more details about the datasets. Like ...

Create date  
Expiry date  
Dataset security  
Dataset organization (PS/PO/IS/DA/VSAM etc)

LISTALC HISTORY  
Gives a list as follows:

```
--DSORG--CREATED--EXPIRES--SECURITY
SYS1.TSO.CLIST
PO      07/01/95 00/00/00   RACF
CATALOG.TCSE.MCAT
**      07/01/89 01/01/99   RPWD
PREFIX.ISPF.PROFILE
PO      07/01/91 01/01/99   NONE
```

## **LISTCAT Operands**

List entries from a catalog.

No operands  
ENTRIES  
ALL  
CATALOG

Catalog stores information about datasets and this information can be obtained by LISTCAT command.

Example: -

### **LISTCAT**

Gives a list of datasets beginning with your Prefix as shown --

```
In CATALOG : CATALOG.TSO1.UCAT
PREFIX.CLIST
PREFIX.CNTL
PREFIX.LOG.MISC
PREFIX.ISR001.BACKUP
```

To List catalog information about specific dataset, use ENTRIES option.

Example: -

### **LISTCAT ENTRIES(CLIST)**

Gives the catalog in which PREFIX.CLIST is present

```
NONVSAM  ----- PREFIX.CLIST
IN-CAT   ----- CATALOG.TSO1.UCAT
```

To List more information about a specific dataset, use ALL option.

Example: -

### **LISTCAT ENT(CLIST) ALL**

Gives the following information for PREFIX.CLIST

```
NONVSAM  ----- PREFIX.CLIST
IN-CAT   ----- CATALOG.TSO1.UCAT
HISTORY
OWNER-IDEN --- (NULL)   CREATION --- 1995.001
VOLUMES
VOLSER   ----- TSO001   DEVTYPE --- x'30102030'
ASSOCIATIONS --- (NULL)
```

To list information from a specific catalog use the CATALOG option and give the name of the catalog.

Example: -

### **LISTCAT CATALOG('CATALOG.APL2.UCAT') OUTFILE(catdd)**

Writes the information in 'apl2.ucat' catalog in the dataset associated with the given ddname.



## **LISTDS Command**

Lists the attributes of a dataset.

Examples: -

```
LISTDS PARTS.DATA
```

```
LISTDS PARTS.DATA STATUS
```

The operand STATUS lists the ddname associated with the dataset and the datasets disposition.

System responds as

```
PREFIX RTS.DATA
```

```
--RECFM-LRECL-BLKSIZE-DSORG-DDNAME-DISP
```

```
VB 255 5100 PO MYFILE KEEP
```

```
--VOLUMES--
```

```
TSO001
```

## **LISTDS MEMBERS operand**

This lists all the members of the dataset.

Example: -

```
LISTDS PARTS.DATA MEMBERS
```

System responds as

```
PREFIX RTS.DATA
```

```
--RECFM-LRECL-BLKSIZE-DSORG-DDNAME-DISP
```

```
VB 255 5100 PO MYFILE KEEP
```

```
--VOLUMES--
```

```
TSO001
```

```
--MEMBERS--
```

```
PART4
```

```
PART5
```

```
PART0
```

## **RENAME Command.**

Can be used for :

- Change the name of -  
Non-VSAM catalogued dataset.  
A member of a catalogued PDS.
- Create an ALIAS for a member of a PDS.

Examples: -

- 1) RENAME PARTS.DATA PART1.DATA
- 2) RENAME \*.TEXT \*.LOAD
- 3) RENAME PARTS.DATA(MEM1) PARTS.DATA(NEW1)
- 4) RENAME PARTS.DATA(MEM1) (ALIASMEM) ALIAS

To Change names of VSAM datasets have to use Access Method Services ALTER Command.

## **SMCOPY Command**

Used to COPY

- Members of a PDS.
- Sequential datasets.
- Sequential datasets as members of a PDS.
- member of a PDS as a Sequential file.

Datasets Copied must have :

- Fixed or Variable format.
- Blocked or Unblocked.
- Max LRECL of 256.

Examples: -

```
SMCOPY FDS(A.DATA(mem1)) TDS(B.DATA(m1)) NOTRANS
```

The option NOTRANS prevents characters from being converted to uppercase.

The Option FDS (FROMDATASET) is used as Input Dataset and option TDS(TODATASET) is the Destination Dataset.

To Copy parts of a dataset :

```
SMCOPY FDS(A.DATA(MEM1)) LINE(1:5) TDS(SEQ.DATA) NOTRANS
```

Copies the first five lines to a sequential dataset.

## ***PRINTDS Command***

Used to Print :

- A sequential dataset
- One or all members of a PDS
- Part of a dataset (LINES option)
- Control length of the printout (FOLD/TRUNC option)

Examples: -

```
PRINTDS DATASET(NEW.DATA)
```

If 'PREFIX.NEW.DATA' is a PDS, all the members + the directory is printed by this command.

To Print only the members of the PDS without the directory

```
PRINTDS DATASET(NEW.DATA) MEMBERS
```

```
PRINTDS DATASET(NEW.DATA(last)) LINES(1:150) COPIES(5)
```

```
PRINTDS DATASET(NEW.DATA(last)) FOLD(80)
```

With the FOLD Option the lines longer than 80 bytes will be printed in one or more subsequent lines.

```
PRINTDS DATASET(NEW.DATA(last)) TRUNCATE(80)
```

## ***DELETE Command***

Used to Delete one or more datasets or members of PDS from the system.

- On deleting a dataset, the catalog entry is removed.
- On deleting a member, the directory of the PDS gets updated.

Examples: -

```
DELETE NEW.DATA/PASSWRD
```

This dataset was password protected.

```
DELETE (NEW.DATA, NEW.PROG)
```

Deletes both the datasets.

```
DELETE NEW.PROG CATALOG(USERCAT1)
```

Deletes the dataset's entry from the catalog.

## **COMMUNICATE with other users in TSO/E**

Following commands are used for sending and receiving messages from other TSO/E users.

SEND	- short messages to other users
TRANSMIT	- send datasets and long messages
LISTBC	- display messages sent by system operator

Examples: -

```
SEND 'Don't forget the meeting at 17:30.' USER (user1,user2) LOGON
```

```
SEND 'Life is wonderful' USER(*)  
This message is sent to your userid.
```

```
LISTBC  
You might see :  
Message from your friendly Operations:  
The System will be down on Friday Evening from 6PM to 8PM.
```

## **TRANSMIT**

Transmit command can be used to :

- send a message
- send dataset
- send selected members of a PDS

Examples: -

```
To transmit a message to USER2 at node XYZAB03:  
TRANSMIT XYZAB03.USER2  
This command allows you to enter the text in screen mode and on pressing PF3 the message is sent.
```

```
To transmit a dataset :  
TRANSMIT XYZAB03.USER2 DATASET(SAM.ABC)
```

## **RECEIVE**

To receive messages and datasets sent to you.

Examples: -

```
RECEIVE  
You will see :
```

```
'Dataset SAM.ABC from USER2 on XYZAB01  
Enter RESTORE parameters or "DELETE" or "END" '
```

```
To receive into a dataset PREFIX.ABC  
Press the ENTER key.  
System replies as follows:
```

```
Restore successful to dataset 'PREFIX.ABC'  
No more files remain for processing.
```

## ***RUNNING/EXECUTING programs***

Before a program can be executed it must be prepared.

- Compile Source code.
- Link edit and load the program(s).

Programs can be run in two ways in TSO/E

- Foreground : issue commands at terminal. (RUN / CALL)
- Background : JCL statements to SUBMIT jobs.

### **Foreground mode --**

To load and execute a program that is in executable form use CALL.

```
CALL 'prefix.myprog.load(hellow)'
```

To pass parameters to the program

```
CALL 'prefix.myprog.load(hellow)' 'myname,addr'
```

## **BACKGROUND**

Following are the commands used to control the processing of batch jobs.

```
SUBMIT - submits the job for processing  
STATUS / CANCEL
```

Examples : -

```
SUBMIT 'PREFIX.CNTL(RUN)'  
This command submits the JCL 'RUN'  
from the dataset 'PREFIX.CNTL'
```

```
SUBMIT (CNTL(JOB1), INPDATA)  
The dataset 'PREFIX.INPDATA' contains  
the input to the job.
```

The command STATUS is used to query the status of the job or userid.  
The command CANCEL is used to cancel a job.

```
CANCEL userid1  
Where userid1 is the job-id.
```

Example:-

Enter STATUS userid at ready prompt...

```
READY  
status btdbks
```

You will see like ...

```
JOB BTDBKS (TSU07304) ON OUTPUT QUEUE  
JOB BTDBKS (TSU08820) ON OUTPUT QUEUE  
JOB BTDBKS (TSU08929) EXECUTING  
READY
```

## TSO Command Summary

ALLOCATE	<ul style="list-style-type: none"> <li>- Open an existing dataset If you Allocate a dataset that exists, the system allows you to open and use the dataset.</li> <li>- Define a new dataset If you Allocate a dataset that is new, the system creates space for it and allows you to open the dataset.</li> </ul>
CANCEL	- is used to cancel a job.
DELETE	- Used to Delete one or more datasets or members of PDS from the system.
FREE	<ul style="list-style-type: none"> <li>- Releases the datasets that were previously allocated in a TSO/E Session.</li> <li>Releases the ddnames that link the datasets to programs.</li> </ul>
LISTALC	- Used to obtain a list of the datasets allocated during the current TSO/E Session.
LISTBC	- Display messages sent by system
LISTCAT Operands	- List entries from a catalog.
LISTDS	- List attribute information about datasets
LISTDS MEMBERS operand	- This lists all the members of the dataset.
PRINTDS	Used to Print : <ul style="list-style-type: none"> <li>- A sequential dataset</li> <li>- One or all members of a PDS</li> <li>- Part of a dataset (LINES option)</li> <li>- Control length of the printout (FOLD/TRUNC option)</li> </ul>
RECEIVE	- To receive messages and datasets sent to you.
RENAME	Change the name of - <ul style="list-style-type: none"> <li>- Non-VSAM catalogued dataset.</li> <li>- A member of a catalogued PDS.</li> <li>Create an ALIAS for a member of a PDS.</li> </ul>
SEND	- Send short messages to other users
SMCOPY	Used to COPY <ul style="list-style-type: none"> <li>- Members of a PDS.</li> <li>- Sequential datasets.</li> <li>- Sequential datasets as members of a PDS.</li> <li>- Member of a PDS as a Sequential file.</li> </ul>
STATUS.	- is used to query the status of the job or userid
SUBMIT	- Submits the job for processing
TRANSMIT	- Send datasets and long messgaes