

**Appendix
B**

**TSO
COMMAND
SUMMARY**

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

1 ALLOCATE Command

For a new data set:

ALLOCATE DSNAME(data-set-name)

 DDNAME(ddname)
 FILE(ddname)

 {KEEP }
 {CATALOG}
 {DELETE }

 UNIT(device)

 VOLUME(volume-serial-number)

 SPACE(primary secondary)
 DIR(directory-space) PDS files only

 {TRACKS }
 {CYLINDERS}

 {EXPDT(expiration-date) }
 {RETPD(retention-period)}

 DSORG(organization)

 RECFM(record-format)

 LRECL(record-length)

 BLKSIZE(block-size)

 LIKE(model-data-set-name)

The ALLOCATE command (continued)

For an existing data set:

```
ALLOCATE      {DSNAME(data-set-name)}  
              {DATASET(data-set-name)}  
  
              {DDNAME(ddname)}  
              {FILE(ddname)}  
  
              {OLD}  
              {SHR}  
              {MOD}  
  
              {KEEP  }  
              {DELETE }  
              {UNCATALOG}  
  
              UNIT (device)  
  
              VOLUME (volume-serial-number)
```

For terminal I/O:

```
ALLOCATE      DSNAME(*)  
  
              DDNAME(ddname)
```

For a dummy file:

```
ALLOCATE      DUMMY  
  
              DDNAME(ddname)
```

The ALLOCATE command (continued)

For SYSOUT output:

ALLOCATE DDNAME(ddname)
 SYSOUT(class)
 {{HOLD }}
 {{NOHOLD}}
 [DEST(station-id)]

CALL program-name [parameter-string']

CANCEL {job-name }
 {job-name(job-id)}
 [PURGE]
 [NOPURGE]

COBOL data-set-name(member)
 {{LOAD(object-module-name)}}
 {{NOLOAD }}
 {PRINT(print-file-name)}
 {{PRINT(*) }}
 {NOPRINT }
 {{LIB(library-names)}}
 {{NOLIB }}
 [compiler-options]

COPY old-data-set-name new-data-set-name

DELETE (data-set-names)
 [PURGE]

DSPRINT data-set-name
 printer-name

 {NUM(location,length) }
 [{{SNUM(location,length)}}]
 {NONUM }

 [LINES(start[:end])]

 {SINGLE}
 [{{DOUBLE}}]
 {CCHAR }

 [{{FOLD }}]
 [{{TRUNCATE}}]

 [{{EJECT }}]
 [{{NOEJECT}}]

The EDIT command

EDIT library(member) type {{OLD}}
 {{NEW}}

EDIT subcommands

CHANGE start-line [end-line] old-string new-string [ALL]

DELETE start-line [end-line]

END {SAVE }
 {NOSAVE}

INPUT line-number [increment]

LIST [start-line] [end-line]

RENUM [new-first-line] [increment]

The EXEC commandExplicit form

EXEC proc-lib-name(proc-name) [parameters'] [control-options]

Implicit form

[%] proc-name [parameters]

FREE {ALL }
{DSNAME(data-set-names)}
{DDNAME(ddnames) }

{KEEP }
[CATALOG]
[UNCATALOG]
{DELETE }

[SYSOUT(class)]

[HOLD]
[NOHOLD]

[DEST(station-id)]

HELP [command-name]

LINK data-set-name
[LOAD(load-module-name)]

{PRINT(print-file-name)}
[PRINT(*)]
{NOPRINT }

[LIB(library-names)]

COLIB

LIST data-set-name

LISTALC [STATUS]
[HISTORY]
[MEMBERS]
[SYSNAMES]

LISTCAT {{ENTRIES(data-set-names)}}
{{LEVEL(level) }}
{NAME }
{{HISTORY}}
{{VOLUME }}
{ALL }

LISTDS (data-set-names)
[MEMBERS]
[HISTORY]
[STATUS]
[LEVEL]

LOADGO data-set-name
[parameter-string]
{PRINT(print-file-name)}
{{PRINT(*) }}
{NOPRINT }
[LIB(library-names)]
COLIB

LOGOFF

LOGON user-id/password ACCT(account-number) [NONOTICE]

OUTPUT {job-name }
{job-name(job-id)}
[NEWCLASS(class-name)]
[DEST(station-id)]
{{HOLD }}
{{NOHOLD}}
[CLASS(class-names)]
[DELETE]
[PAUSE]

OUTPUT subcommands:

CONTINUE {BEGIN}
 {{HERE }}
 {NEXT }

 {{PAUSE }}
 {{NOPAUSE}}

END

SAVE data-set-name

RENAME old-name new-name

[ALIAS]

STATUS {{job-name }}
 {{job-name(job-id)}}

SUBMIT data-set-name
 [JOBCHAR(character)]

TESTCOB (program-id:ddname1)
LOAD(member:ddname2)
[PRINT(ddname3)]
[PARM(`parameter-string`)]

TESTCOB subcommands:

AT statement-list
[(subcommand-list)]

DROP [symbol] Use with EQUATE subcommand

DUMP

END

EQUATE symbol data-name

GO [program-id] [.statement-number] [.verb-number]

IF (expression) {GO }
{HALT}

LIST {identifier-list}
 {ALL }

 [PRINT]

LISTFILE file-name

 [PRINT]

LSTBRKS

NEXT

OFF [statement-list] (Use with AT subcommand)

OFFWN [identifier] (Use with WHEN subcommand)

RUN [program-id] [.statement-number] [.verb-number]

SET identifier-1 = {identifier-2}
 {literal }

SOURCE {line-1 } ddname
 {line-1:line-2}

 {ENTRY}
 [{PARA }]

TRACE [**{NAME }**]
 {OFF }

 [**PRINT**]

WHEN identifier **{data-name }**
 {(expression)}

