

# SHARE PROGRAM LIBRARY AGENCY



PROGRAM NUMBER

**036027**

## University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA  
(305) - 284-6257

# SHARE PROGRAM LIBRARY SUBMITTAL FORM



SHARE PROGRAM LIBRARY AGENCY  
Triangle Universities Computation Center  
Post Office Box 12076  
Research Triangle Park, North Carolina USA 27709

SPLA

CONTROL NUMBER:

This form should be completed and submitted with the program package to the SHARE Program Library Agency at the address shown above. Standards and instructions for submitting programs are in the SHARE Reference Manual, Section 6.

- (1) Program Number (to be filled by SPLA) . . . . . 360D - 03.6.027  
Time Sharing Language/One (TL/I)
- (2) Title of Program . . . . .  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- (3) System Type(s) (Machine). . . . . MVS Release 3.0 or later  
Command Language, TSO, CLIST, TLI
- (4) Search Key(s) . . . . .  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_
- (5) Programming Systems/Languages . . . . . Assembler & PL/I (Optimizer)
- (6) Primary Subject Code . . . . . MVS/TSO; MVS Release 3.0, SU's 4,5, and 7
- (7) Minimum System Requirements \_\_\_\_\_
- (8) New (N) or Revision (R) (if revision, show prior Program Number in Item 1) R
- (9) Date of Submittal . . . . . 9/5/80
- (10) Documentation (number of original pages submitted) . . . . . \_\_\_\_\_
- (11) Author's Name and Address . . . . . Software Development  
Standard Oil Company (Indiana)  
200 East Randolph Drive  
Chicago, Illinois 60601
- (12) Direct Technical Inquiries to Name & Address (if different than Author) \_\_\_\_\_  
Perry Sugerman  
Corporate Software Support  
Standard Oil Co. (Indiana)  
Mail Code 0404  
200 E. Randolph Dr.  
Chicago, IL 60601  
(312) 856-5435  
IN
- (13) Submitter's Installation Membership Code . . . . . \_\_\_\_\_
- (14) Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

## DISCLAIMER

Triangle Universities Computation Center (TUGC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUGC nor SHARE, INC. make any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

## SHARE PROGRAM LIBRARY SUBMITTAL FORM

**Subject Guide:**

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements

**ABSTRACT --** Time Sharing Language/One is a TSO Command Processor Language based around PL/I. It can be used for highly specialized interactive applications. When used as a CLIST replacement, it provides the user with all the logical power available to the PL/I programmer. Time Sharing Language/One is distributed as PL/I Pre-Processor Macros, PL/I Subprograms, Assembly Language Subprograms, and Sample Commands.

**DISCLAIMER**  
Triangle Universities Computation Center (TUCC) serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, Inc., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

(Please attach additional pages if necessary) . . . . . Total pages attached \_\_\_\_\_

An "Acknowledgement of Assistance" statement must be attached to this Submittal Form.

**Permission to Publish**

"I hereby give the SHARE Program Library Agency permission to reprint, reproduce, and distribute this program"

(15) Signature of Submitter and Date

*Stephen D. Strickland* 9/5/80

(15) Signature of Installation Addressee

*Perry Saperman (LMP)* 9/5/80

INSTALLATION NOTES FOR  
TIME SHARING LANGUAGE/ONE

The Time Sharing Language One (TL/I) Installation Tape is a standard label tape (VOL=SER=CI5470) containing six (6) files, all in IEBCOPY unload format. These are:

- File 1: TL/I ASM Source Modules  
(DSN=CSD.SHARE.TLI.ASM 60 tracks)
- File 2: TL/I PL/I Source Modules  
(DSN=CSD.SHARE.TLI.PLI1 10 tracks)
- File 3: TL/I Sample Command PL/I Source Modules  
(DSN=CSD.SHARE.TLI.PLI2 20 tracks)
- File 4: PL/I Optimizer and BAL Macros  
(DSN=CSD.SHARE.TLI.MACRO 40 tracks)
- File 5: TSO Help Entries for Sample Commands  
(DSN=CSD.SHARE.TLI.HELP 10 tracks)
- File 6: TL/I Link Decks  
(DSN=CSD.SHARE.TLI.LINK 10 tracks)

The first step in installing TL/I is to allocate libraries corresponding to these six files and load them using IEBCOPY. Suggested track allocations given above are calculated for 3350 disk drives.

Secondly, the TL/I source must be assembled/compiled. Note that all link decks expect to include all modules from a library named 'SYSU.UNMVS'. If you choose to have all modules in NCAL load module form, then 'SYSU.UNMVS' should be your NCAL load library. Otherwise, 'SYSU.UNMVS' should be allocated as your object library. For correct compilation your PL/I job stream must declare the macro dataset on its SYSLIB statement. For correct assemblies, your BAL job stream will need a SYSLIB concatenation including the TL/I macro dataset, 'SYS1.MACLIB', 'SYS1.AMODGEN', and 'SYS1.ATSOMAC'.

At this point the modules may be linked. Note that one of the link decks uses a procedure called LINKS to execute the linkage editor. This procedure has been included in the link deck dataset and should be moved to a procedure library. Output load modules from the link decks are directed to three authorized libraries: 'SYS1.CMDLIB', 'SYS1.LPALIB', and 'U.ULMVS'. Note that a few of the command link decks require modules which were not included in this shipment. These command source modules and their link decks were provided purely as samples.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-2

Section 24

Page  
1

Section Name TSO UTILITIES

Subsection Name General

Subject

The TSO Utilities are a set of TSO command processors and the modules which support them, written either in TL/I (a high-level PL/I-based command language for TSO) or in Assembly Language.

TL/I was developed to provide a high-level, noninterpretative, user-oriented, easily-interfaced language for writing TSO command processors. For documentation of the language itself, see the TL/I Reference Manual, Programmer's Guide, and Program Logic/Data Structures Manuals. The set of commands based on this language and currently available is described under 'TSO Utilities' in FSD manuals.



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.1	Page	1
Section Name TSO UTILITIES					
Subsection Name Distribution Materials Description					
Subject Source Modules, Tables, Macros					

Date issued 01-18-78

Replaces Issue

Following is a list of all \$ commands and support routines by dataset in which they are located:

Source (commands) in SYSU.USMVS

\$  
\$CALL  
\$CATFI  
\$COPY  
\$DBALLOC  
\$DBFREE  
\$DDALLOC  
\$PDS  
\$PMAP  
\$PRINT  
\$RESET  
\$TR  
\$WHO

Source (Support Routines) in SYSU.USMVS

TLIACCT  
TLIARRY  
TLIASR  
TLIATTN  
TLICALL  
TLICAPS  
TLICAT  
TLICMD  
TLICP  
TLICOPY  
TLIDAIR  
TLIDCAT  
TLIDDN  
TLIDEL  
TLIDRC  
TLIDSN  
TLIDSX  
TLIENQ#  
TLIENQ2#  
TLIGET  
TLIHEX  
TLIKEYS  
TLILINK  
TLILNK



Standard Oil  
Company (Indiana)

Date Issued

01-18-78

Replaces Issue

Manual

FSD-2

Section

24.1

Page  
2

Section Name

TSO UTILITIES

Subsection Name

Distribution Materials Description

Subject

Source Modules, Tables, Macros

TLILOAD  
TLIMEM  
TLINIT  
TLIPDQ  
TLIPDSN  
TLIPUT  
TLISTAI  
TLISTAK  
TLISTOR  
TLITIME  
TLIUSTAK  
TLIXIT  
CATCK

#### Tables

TLIEFTE2  
TLIEFTE8  
IKJEFTE2  
IKJEFTE8

#### Macros in SYSU.UMMVS

TLI\$DDN  
TLICBUF  
TLICOPY  
TLICPPL  
TLIDATA  
TLIDSCB1  
TLIDSCB2  
TLIDSCB3  
TLIDSCB4  
TLIDSA  
TLIECT  
TLIMACRO  
TLIPSCB  
TLIREGS  
TLISTART  
TLISTAX  
TLISTOP  
TLITSB  
TLIUPT



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
1

Section Name TSO UTILITIES

Date Issued 01-18-78

Subsection Name Installation JCL

Replaces Issue

Subject

Link JCL for the \$ commands includes the following (described below):

1. JCL to relink TMP
2. Special PL/I prelink
3. Link decks for \$ commands

#### Relink TMP

If tables IKJEFTE2 (APF command table) and IKJEFTE8 (APF program table) are being changed, the TMP must be relinked using the following JCL:

```
//STEP7 EXEC LINKS,N=SYS1,NAME=LPALIB,  
//          PARM='RENT,REUS,REFR,MAP,LIST,LET,XREF'  
//UNMVS DD DSN=SYSU.UNMVS,DISP=SHR  
//AOST4 DD DSN=SYS1.AOST4,DISP=SHR  
//SYSLIN DD *  
    REPLACE IKJEFTB2(PCFTMPB2),IKJEFTA2(PCFTMPA2)  
    INCLUDE AOST4(IKJEFT02)  
    INCLUDE AOST4(PCFTMP02)  
    INCLUDE AOST4(IKJEFT03)  
    INCLUDE AOST4(IKJEFT06)  
    INCLUDE UNMVS(IKJEFTE2)  
    INCLUDE UNMVS(IKJEFTE8)  
    ALIAS IKJEFTOB  
    ENTRY IKJEFT02  
    NAME IKJEFT02(R)
```

Note: These two tables would probably only be changed to authorize a new entry point to the \$ command processor. Currently '\$' and '@' are authorized; '\$X' is not and is not in IKJEFTE2.





Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.2	Page 2
Section Name	TSO UTILITIES			
Subsection Name	Installation JCL			
Subject	PL/I Prelink			

Date issued 01-18-78

Replaces Issue

### PL/I Prelink

TL/I support routines are linked into U.ULMVS, and commands into SYS1.CMDLIB, using the following JCL:

```
//STEP3 EXEC PGM=IEWL,PARM='AC=1,LIST,RENT,MAP,XREF,LET,REUS'  
// REGION=512K  
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA  
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR  
//SYSPRINT DD SYSOUT=A  
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS  
//UNUTIL DD DISP=SHR,DSN=SYSU.UNUTIL  
//OBJECT DD DSN=SDDSI.A01717.SM00158,DISP=SHR  
//SYSLMOD DD DSN=U.ULMVS,DISP=SHR  
//SYSLIN DD *  
  INCLUDE OBJECT(TLIARRY)  
  INCLUDE OBJECT(TLIPUT)  
  ENTRY PLISTART  
  NAME TLIARRY(R)  
  INCLUDE OBJECT(TLIATTN)  
  NAME TLIATTN(R)  
  INCLUDE OBJECT(TLICALL)  
  INCLUDE OBJECT(TLISTAI)  
  ENTRY TLICALL  
  NAME TLICALL(R)  
  INCLUDE OBJECT(TLICAPS)  
  NAME TLICAPS(R)  
  INCLUDE OBJECT(TLICAT)  
  NAME TLICAT(R)  
  INCLUDE OBJECT(TLICP)  
  INCLUDE OBJECT(TLISTAI)  
  ENTRY TLICP  
  NAME TLICP(R)  
  INCLUDE OBJECT(TLIDCAT)  
  NAME TLIDCAT(R)  
  INCLUDE OBJECT(TLIDEL)  
  NAME TLIDEL(R)  
  INCLUDE OBJECT(TLIDSN)  
  NAME TLIDSN(R)  
  INCLUDE OBJECT(TLIDSX)  
  NAME TLIDSX(R)  
  INCLUDE OBJECT(TLIGET)  
  NAME TLIGET(R)  
  INCLUDE OBJECT(TLIHEX)  
  NAME TLIHEX(R)  
  INCLUDE OBJECT(TLILINK)  
  NAME TLILINK(R)  
  INCLUDE OBJECT(TLILNK)  
  INCLUDE OBJECT(TLISTAI)  
  ENTRY TLILNK
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
3

Date Issued 01-18-78

Section Name TSO UTILITIES

Subsection Name Installation JCL

Replaces Issue

Subject PL/I PRELINK

```
NAME TLILNK(R)
INCLUDE OBJECT(TLIMEM)
NAME TLIMEM(R)
INCLUDE OBJECT(TLILOAD)
NAME TLILOAD(R)
INCLUDE OBJECT(TLINIT)      /* EXPECT FOLLOWING */
INCLUDE UNUTIL(ABEND)      /* IEW0132 ATTN,RC,STAI */
INCLUDE OBJECT(TLIATTN)
ENTRY TLINIT
NAME TLINIT(R)
INCLUDE OBJECT(TLIPDQ)
NAME TLIPDQ(R)
INCLUDE OBJECT(TILPDSN)
NAME TILPDSN(R)
INCLUDE OBJECT(TLIPUT)
NAME TLIPUT(R)
INCLUDE OBJECT(TLISTAI)
NAME TLISTAI(R)
INCLUDE OBJECT(TLISTAK)
NAME TLISTAK(R)
INCLUDE OBJECT(TLISTOR)
NAME TLISTOR(R)
INCLUDE OBJECT(TLITIME)
NAME TLITIME(R)
INCLUDE OBJECT(TLIUSTAK)
NAME TLIUSTAK(R)
INCLUDE OBJECT(TLIXIT)
NAME TLIIXIT(R)
INCLUDE OBJECT(TLIENQ#)
INCLUDE OBJECT(TLIENQ2#)
NAME TLIENQ#(R)
INCLUDE OBJECT(TLIENQ2#)
NAME TLIENQ2#(R)
INCLUDE OBJECT(TLIKEYS)
INCLUDE OBJECT(TLIPUT)
NAME TLIKEYS(R)
//STEP4 EXEC PGM=IEWL,PARM='AC=1,LIST,MAP,XREF,LET',
//      REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//OBJECT DD DSN=SDDSI.A01717.SM00158,DISP=SHR
//SYSLMOD DD DSN=U.ULMVS,DISP=SHR
```



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.2	Page	4
Section Name		TSO UTILITIES			
Subsection Name		Installation JCL			
Subject		PL/I Prelink			

Date Issued 01-18-78

Replaces Issue

```
//SYSLIN DD *
  INCLUDE OBJECT(TLICOPY)
  INCLUDE OBJECT(TLIXIT)
  ENTRY TLICOPY
  NAME TLICOPY(R)
//STEP5 EXEC PGM=IEWL,PARM='AC=1,LIST,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//OBJECT DD DSN=SDDSI.A01717.SM00158,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
  INCLUDE OBJECT($PRINT)
  INCLUDE OBJECT(TLIATTN)
  INCLUDE OBJECT(TLIXIT)
  INCLUDE OBJECT(TLISTAI)
  INCLUDE OBJECT(TLINIT)
  INCLUDE OBJECT(TLIHEX)
  INCLUDE OBJECT(TLIPUT)
  INCLUDE OBJECT(TLICAPS)
  INCLUDE OBJECT(TLIARRY)
  INCLUDE OBJECT(TLIGET)
  INCLUDE OBJECT(TLITIME)
  INCLUDE OBJECT(TLICP)
  INCLUDE OBJECT(TLILNK)
  INCLUDE OBJECT(TLICOPY)
  ENTRY PLISTART
  ALIAS $PUNCH
  ALIAS $SEND
  NAME $PRINT(R)
  INCLUDE OBJECT($CALL)
  INCLUDE OBJECT(TLISTAI)
  INCLUDE OBJECT(TLIATTN)
  INCLUDE OBJECT(TLICALL)
  INCLUDE OBJECT(TLINIT)
  INCLUDE OBJECT(TLIHEX)
  INCLUDE OBJECT(TLIPUT)
  INCLUDE OBJECT(TLICAPS)
  INCLUDE OBJECT(TLIGET)
  INCLUDE OBJECT(TLIKEYS)
  INCLUDE OBJECT(TLICP)
  INCLUDE OBJECT(TLITIME)
  ENTRY PLISTART
  NAME $CALL(R)
```



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.2	Page 5
Section Name TSO UTILITIES				
Subsection Name Installation JCL				
Subject PL/I Prelink				

Date Issued 01-18-78

Replaces Issue

```
INCLUDE OBJECT($CATFI)
INCLUDE OBJECT(TLIATTN)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLIARRY)
INCLUDE OBJECT(TLIGET)
INCLUDE OBJECT(TLICAT)
ENTRY PLISTART
NAME $CATFI(R)
INCLUDE OBJECT($DDALLOC)
INCLUDE OBJECT(TLIATTN)
INCLUDE OBJECT(TLISTAI)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLIGET)
INCLUDE OBJECT(TLICP)
ENTRY PLISTART
NAME $DDALLOC(R)
INCLUDE OBJECT($PDS)
INCLUDE OBJECT(TLIATTN)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLIPDQ)
INCLUDE OBJECT(TLITIME)
ENTRY PLISTART
NAME $PDS(R)
INCLUDE OBJECT($)
ALIAS E$
ALIAS E@
ALIAS $X
ALIAS E$X
ALIAS @
ENTRY $
INCLUDE OBJECT(TLIEFTE2)
INCLUDE OBJECT(TLIEFTE8)
NAME $(R)
INCLUDE UNMVS(CATCK)
NAME CATCK(R)
INCLUDE OBJECT ($DEGAS)          /* WAS $COPY */
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
6

Section Name TSO UTILITIES

Date Issued 01-18-78

Subsection Name Installation JCL

Replaces Issue

Subject PL/I Prelink

```
INCLUDE OBJECT(TLIKEYS)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLIARRY)
INCLUDE OBJECT(TLIGET)
INCLUDE OBJECT(TLICP)
INCLUDE OBJECT(TLICALL)
INCLUDE OBJECT(TLITIME)
INCLUDE OBJECT(TLISTAI)
INCLUDE OBJECT(TLIATTN)
INCLUDE OBJECT(TLIDSX)
INCLUDE OBJECT(TLIDSN)
INCLUDE OBJECT(TLINEM)
ALIAS $COPY
NAME $DEGAS(R)
INCL'DE OBJECT($WHO)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLIENQ#)
INCLUDE OBJECT(TLITIME)
INCLUDE OBJECT(TLIENQ2#)
INCLUDE OBJECT(TLIATTN)
NAME $WHO(R)
INCLUDE OBJECT($STR)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLICP)
INCLUDE OBJECT(TLIATTN)
INCLUDE OBJECT(TLISTAI)
INCLUDE OBJECT(TLITIME)
NAME $STR(R)
INCLUDE OBJECT($RESET)
INCLUDE OBJECT(TLIATTN)
INCLUDE OBJECT(TLIARRY)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLICP)
INCLUDE OBJECT(TLIGET)
INCLUDE OBJECT(TLIHEX)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLISTAI)
INCLUDE OBJECT(TLITIME)
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
7

Date Issued 01-18-78

Section Name TSO UTILITIES

Subsection Name Installation JCL

Replaces Issue

Subject PL/I Prelink

```
INCLUDE OBJECT(TLIKEYS)
NAME $RESET(R)
//STEP6 EXEC PGM=IEWL,PARM='AC=1,LIST,MAP,XREF,LET',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//OBJECT DD DSN=SDDSI.A01717.SM00158,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE OBJECT($PMAP)
INCLUDE OBJECT(TLINIT)
INCLUDE OBJECT(TLIHGX)
INCLUDE OBJECT(TLIPUT)
INCLUDE OBJECT(TLICAPS)
INCLUDE OBJECT(TLIARRY)
INCLUDE OBJECT(TLIDSX)
INCLUDE OBJECT(TLITIME)
INCLUDE OBJECT(TLIATTN)
NAME $PMAP(R)
//
//
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
8

Section Name TSO UTILITIES

Date Issued 7-21-78

Subsection Name Installation JCL

Replaces Issue 01-18-78

Subject Link Decks

### Link Decks

Following are all link deck listings for the TSO Utilities:

```
//STEP3 EXEC PGM=TEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DISP=SHR,DSN=SYSU.UNUTIL
//SYSLMOD DD DSN=U,ULMVS,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(TLIARRY)
INCLUDE UNMVS(TLIPUT)
ENTRY PLISTART
NAME TLIARRY(R)
INCLUDE UNMVS(TLIATTN)
NAME TLIATTN(R)
INCLUDE UNMVS(TLICALL)
INCLUDE UNMVS(TLISTAI)
ENTRY TLICALL
NAME TLICALL(R)
INCLUDE UNMVS(TLICAPS)
NAME TLICAPS(R)
INCLUDE UNMVS(TLICAT)
NAME TLICAT(R)
INCLUDE UNMVS(TLICP)
INCLUDE UNMVS(TLISTAI)
ENTRY TLICP
NAME TLICP(R)
INCLUDE UNMVS(TLIDCAT)
NAME TLIDCAT(R)
INCLUDE UNMVS(TLIDEL)
NAME TLIDEL(R)
INCLUDE UNMVS(TLIDSN)
NAME TLIDSN(R)
INCLUDE UNMVS(TLIGET)
NAME TLIGET(R)
INCLUDE UNMVS(TLIHGX)
NAME TLIHGX(R)
INCLUDE UNMVS(TLILINK)
NAME TLILINK(R)
INCLUDE UNMVS(TLILNK)
INCLUDE UNMVS(TLISTAI)
ENTRY TLILNK
NAME TLILNK(R)
INCLUDE UNMVS(TLILOAD)
NAME TLILOAD(R)
INCLUDE UNMVS(TLINIT)
INCLUDE UNUTIL(ABEND)
INCLUDE UNMVS(TLIATTN)
ENTRY TLINIT
NAME TLINIT(R)
INCLUDE UNMVS(TLIMEM)
NAME TLIMEM(R)
INCLUDE UNMVS(TLIPDG)
NAME TLIPDG(R)

/* EXPECT FOLLOWING */
/* 15W0132 ATTN.RC,STAI */

** GROUP 3 TSO UTILITIES
** 12/14/77
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
9

Date Issued 7-21-78

Replaces Issue 01-18-78

Section Name TSO UTILITIES

Subsection Name Installation JCL

Subject Link Decks

```
INCLUDE UNMVS(TLIPDSN)
NAME TLIPDSN(R)
INCLUDE UNMVS(TLIPUT)
NAME TLIPUT(R)
INCLUDE UNMVS(TLISTAI)
NAME TLISTAI(R)
INCLUDE UNMVS(TLISTAK)
NAME TLISTAK(R)
INCLUDE UNMVS(TLISTOR)
NAME TLISTOR(R)
INCLUDE UNMVS(TLITIME)
NAME TLITIME(R)
INCLUDE UNMVS(TLIUSTAK)
NAME TLIUSTAK(R)
INCLUDE UNMVS(TLIXIT)
NAME TLIXIT(R)
INCLUDE UNMVS(TLIENQ#)
INCLUDE UNMVS(TLIENQ2#)
NAME TLIENQ#(R)
INCLUDE UNMVS(TLIENQ2#)
NAME TLIENQ2#(R)
INCLUDE UNMVS(TLIKEYS)
INCLUDE UNMVS(TLIPUT)
NAME TLIKEYS(R)
```

```
/*
//STEP4 EXEC PGM=IEWL,PARM='AC=1,CALL,MAP,XREF,LET',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1,PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU,UNUTIL,DISP=SHR
//SYSLMOD DD DSN=U,ULMVS,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(TLICOPY)
INCLUDE UNMVS(TLIXIT)
ENTRY TLICOPY
NAME TLICOPY(R)
INCLUDE UNMVS(TLIDSX)
NAME TLIDSX(R)
```

```
/*
//STEPS EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1,PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU,UNUTIL,DISP=SHR
//ULUTIL DD DSN=U,ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1,CNOLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(CATCH)
NAME CATCH(R)
```





Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
10

Section Name TSO UTILITIES

Date Issued 7-21-78

Subsection Name Installation JCL

Replaces Issue 01-18-78

Subject Link Decks

```
//* *****  
//* *****  
//* RUN THE FOLLOWING STEP IF PCF IS INSTALLED ON YOUR SYSTEM.  
//* *****  
//* *****  
//STEP7 EXEC LINKS,N=SYS1,NAME=LPALIB,  
//          PARM=RENT,REUS,REFR,MAP,LIST,LET,XREF'  
//UNMVS DD DSN=SYSU,UNMVS  
//AOST4 DD DSN=SYS1,AOST4,DISP=SHR  
//SYSLIN DD *  
//      REPLACE IKJEFTB2(PCFTMPB2),IKJEFTA2(PCFTMPA2)  
//      INCLUDE AOST4(IKJEFT02)  
//      INCLUDE AOST4(PCFTMP02)  
//      INCLUDE AOST4(IKJEFT03)  
//      INCLUDE AOST4(IKJEFT06)  
//      INCLUDE UNMVS(IKJEFT02)  
//      INCLUDE UNMVS(IKJEFT08)  
//      ALIAS IKJEFT08  
//      ENTRY IKJEFT02  
//      NAME IKJEFT02(R)
```

```
//STEP7 EXEC LINKS,N=SYS1,NAME=LPALIB,  
//          PARM=RENT,REUS,REFR,MAP,LIST,LET,XREF'  
//UNMVS DD DSN=SYSU,UNMVS,DISP=SHR  
//AOST4 DD DSN=SYS1,AOST4,DISP=SHR  
//SYSLIN DD *  
//      INCLUDE AOST4(IKJEFT02)  
//      INCLUDE AOST4(IKJEFT03)  
//      INCLUDE AOST4(IKJEFT06)  
//      INCLUDE UNMVS(IKJEFT02)  
//      INCLUDE UNMVS(IKJEFT08)  
//      ORDER IKJEFT02(P)  
//      ORDER IKJEFT03  
//      ORDER IKJEFT06  
//      ENTRY IKJEFT02  
//      ALIAS IKJEFT03  
//      ALIAS IKJEFT06  
//      NAME IKJEFT02(R)
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
11

Date Issued 7-21-78

Section Name TSO UTILITIES

Subsection Name Installation JCL

Replaces Issue 01-18-78

Subject Link Decks

```
//STEPS EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS($)
ALIAS ES
ALIAS E#
ALIAS EX
ALIAS ESX
ALIAS #
ALIAS $
INCLUDE UNMVS(TLIEFTE2)
INCLUDE UNMVS(TLIEFTE8)
NAME $CR)
```

```
//STEPS EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SCATFI)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHFX)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIARRY)
INCLUDE UNMVS(TLIGET)
INCLUDE UNMVS(TLICAT)
ENTRY PLISTANT
NAME $CATFI(R)
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
12

Section Name TSO UTILITIES

Date Issued 7-21-78

Subsection Name Installation JCL

Replaces Issue 01-18-78

Subject Link Decks

```
//STEP2 EXEC PGM=IEWL,PARM='AC=1,NCAL,MAP,XREF,LET',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIN DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DSN=SYSU.UNMVS,DISP=SHR
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SDBALLOC)
INCLUDE UNMVS(TLISTAT)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHFX)
INCLUDE UNMVS(TLIPTUT)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIGET)
INCLUDE UNMVS(TLICP)
INCLUDE UNUTIL(CALLED)
ENTRY PLISTART
NAME SDBALLOC(R)
```

```
/*
//STEP1 EXEC PGM=IEWL,PARM='AC=1,NCAL,MAP,XREF,LET',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//UNMVS DD DSN=SYSU.UNMVS,DISP=SHR
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SDBFREE)
INCLUDE UNMVS(TLISTAT)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHFX)
INCLUDE UNMVS(TLIPTUT)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIGET)
INCLUDE UNMVS(TLICP)
INCLUDE UNUTIL(CALLED)
ENTRY PLISTART
NAME SDBFREE(R)
```

\*\*ADDED 6/20/78

```
//STEP5 EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIN DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SDBALLOC)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLISTAT)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHFX)
INCLUDE UNMVS(TLIPTUT)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIGET)
INCLUDE UNMVS(TLICP)
ENTRY PLISTART
NAME SDBALLOC(R)
```



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
13

Section Name TSO UTILITIES

Date Issued 7-21-78

Subsection Name Installation JCL

Replaces Issue 01-18-78

Subject Link Decks

```
//STEP6 EXEC PGM=IEWL,PARM='AC=1,CALL,MAP,XREF,LET',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS($DEGAS)
INCLUDE UNMVS(TLIKEYS)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHFX)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIARRY)
INCLUDE UNMVS(TLIGET)
INCLUDE UNMVS(TLICP)
INCLUDE UNMVS(TLICALL)
INCLUDE UNMVS(TLITIME)
INCLUDE UNMVS(TLISTAT)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLIDSX)
INCLUDE UNMVS(TLIDSN)
INCLUDE UNMVS(TLIMEM)
ALIAS $COPY
NAME $DEGAS(R)
```

\*\* ADDED AS PART OF GROUP 3  
\*\* TSO UTILITIES  
\*\* 12/14/77

```
//STEP5 EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SPOS)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHFX)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIPDG)
INCLUDE UNMVS(TLITIME)
ENTRY PLISTART
NAME SPOS(R)
```



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.2	Page	14
Section Name		TSO UTILITIES			
Subsection Name		Installataon JCL			
Subject		Link Decks			

Date issued 7-21-78

Replaces Issue 01-18-78

```
//STEP6 EXEC PGM=IEWL,PARM='AC=1,CALL,MAP,XREF,LET',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SPMAP)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHIX)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIARRY)
INCLUDE UNMVS(TLITIME)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLIDSX)
NAME SPMAP(R)
```

```
//STEPS EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS(SPRINT)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLIXIT)
INCLUDE UNMVS(TLISTAI)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHIX)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIARRY)
INCLUDE UNMVS(TLITIME)
INCLUDE UNMVS(TLICP)
INCLUDE UNMVS(TLILNK)
INCLUDE UNMVS(TLIGET)
ALIAS BRUNCH
ALIAS SEND
ENTRY PISTART
NAME SPRINT(R)
```

\*\* 12 /14/77



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.2

Page  
15

Section Name TSO UTILITIES

Date Issued 7-21-78

Subsection Name Installation JCL

Replaces Issue 01-18-78

Subject Link Decks

```
//STEPS EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS($TR)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHEX)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLICP)
INCLUDE UNMVS(TLIGTAT)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLITIME)
NAME $TR(R)
```

\*\*ADDED 6/20/78

```
//STEPS EXEC PGM=IEWL,PARM='AC=1,CALL,RENT,MAP,XREF,LET,REUS',
// REGION=512K
//SYSUT1 DD SPACE=(CYL,(3)),UNIT=SYSDA
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SYSPRINT DD SYSOUT=A
//UNMVS DD DISP=SHR,DSN=SYSU.UNMVS
//UNUTIL DD DSN=SYSU.UNUTIL,DISP=SHR
//ULUTIL DD DSN=U.ULUTIL,DISP=SHR
//SYSLMOD DD DSN=SYS1.CMDLIB,DISP=SHR
//SYSLIN DD *
INCLUDE UNMVS($WHO)
INCLUDE UNMVS(TLIPUT)
INCLUDE UNMVS(TLINIT)
INCLUDE UNMVS(TLIHEX)
INCLUDE UNMVS(TLICAPS)
INCLUDE UNMVS(TLIENQ#)
INCLUDE UNMVS(TLITIME)
INCLUDE UNMVS(TLIATTN)
INCLUDE UNMVS(TLIENQ2#)
NAME $WHO(R)
```



Standard Oil  
Company (Indiana)

Date issued 01-18-78

Replaces issue

Manual	FSD-2	Section	24.3	Page	1
Section Name	TSO UTILITIES				
Subsection Name	Datasets				
Subject					

Macros in SYSU.UMMVS  
Source in SYSU.USMVS  
Load modules in U.ULMVS and SYS1.CMDLIB  
Link decks in SYSU.UDMVS



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual	FSD-2	Section	24.4	Page 1
Section Name TSO UTILITIES				
Subsection Name SYSGEN Parameters				
Subject				

None





Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-2

Section 24.5

Page  
1

Section Name TSO UTILITIES

Subsection Name Testing Guide

Subject

Following is a list of Installation Verification Procedures for the \$ commands (TSO Utilities):

\$CALL ENTER: (after allocating appropriate datasets)  
\$ CALL IEUASM

VERIFY: IEUASM should be found in system library and called.

&COPY ENTER: \$ COPY dataset1 dataset2  
(where dataset1 is to be copied to dataset2)

VERIFY: Check that dataset has been copied.

\$DBALLOC  
\$DBFREE

This IVP must be run on a machine which has the TSO command enhancements package installed. If not, then TSO must be initialized, and these test commands must be entered by hand and verified.

After each of the commands have been executed, the following messages should be produced:

IEC130I ARMSPPDS DD STATEMENT MISSING  
DD-CARD FOR ARMSPPDS MISSING

Any messages produced other than the above, suggest that these components have not been properly installed.

```
//userid EXEC PGM=IKJEFT01,DYNAMNBR=10,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSTSPRT DD SYSOUT=A
//SYSUADN DD DSN=SYS1.UADS,DISP=SHR
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC DD DSN=SYS1.BROADCAST,DISP=SHR
//SYSTSIN DD *
$ DBALLOC (//ANYOLCRD PSB)
$ DBFREE (//ANYOLCRD PSB)
/*
```

\$DDALLOC ENTER: \$DDALLOC (//SOURCE DD DSN=SYS1.HELP,DISP=SHR)

VERIFY: Use the LISTA command to verify that an allocate command has been generated for this DD statement.



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.5

Page  
2

Section Name TSO UTILITIES

Subsection Name Testing Guide

Date Issued 01-18-78

Replaces Issue

Subject

\$DEGAS

ENTER: \$ DEGAS PROGRAMS.ASM

VERIFY: Check to see that your PROGRAMS.ASM was successfully compressed.

\$PDS

ENTER: \$ PDS PROGRAMS.LOAD

VERIFY: When \$PDS displays 'ENTER OPTIONS' try the various options as indicated in the HELP entry for \$PDS.

\$PMAP

ENTER: \$ PMAP PROGRAMS.ASM

VERIFY: Should display attribute information associated with your PROGRAMS.ASM.

\$PRINT

ENTER: \$ PRINT PROGRAMS.ASM(member)

VERIFY: Check that member has been printed.

\$PUNCH

ENTER: \$ PUNCH PROGRAMS.ASM(member)

VERIFY: Check that member has been punched.

\$SEND

ENTER: \$ SEND PROGRAMS.ASM(member)

VERIFY: Check that member has been printed.

\$TR

ENTER: \$ TR followed by transmit operands.

VERIFY: Check that transmit operation completes successfully.

\$WHO

ENTER: \$ WHO 'SYS1.HELP'

VERIFY: Should display all jobs/users enqueued on SYS1.HELP.



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.6	Page	1
Section Name	TSO UTILITIES				
Subsection Name	Special Considerations				
Subject					

Date Issued 1-25-78

Replaces Issue 1-18-78

#### Note 1:

Some \$ commands must be linked as re-entrant; others cannot be re-entrant. (See listing of link decks for specific members.) Note only re-entrant modules can be linked into LPALIB.

#### Note 2:

The TSO Utilities utilize the Authorized Program Facility (APF) to restrict the use of certain programs and services through TSO. Following is a description of the use of APF by the TSO utilities, and rules which consequently must be followed to ensure the TL/1 commands run correctly.

The TMP uses table IKJEFTE2 to decide whether to attach a command processor RSAPF=YES or NO. (The TMP at this point is not running APF authorized.)

1. '\$' is in the table and is attached RSAPF=YES.  
Because \$ resides in an authorized library and is linked AC=1, it gains APF authorization and runs authorized.

2. '\$X' is not in the table, is attached RSAPF=NO, and runs unauthorized.

\$ (and \$X) attach all \$ commands RSAPF=NO. All commands invoked with entry '\$' must be in authorized libraries, and will run authorized. Commands invoked with '\$X' do not run authorized, and therefore, cannot request restricted services.

Command \$CALL (running authorized if invoked with '\$', unauthorized if invoked with '\$X') may attach a requested program RSAPF=YES or NO as follows:

1. If \$CALL is running unauthorized, it attaches RSAPF=NO, and the attached program runs unauthorized.
2. IF \$CALL is running authorized, it tries to find the name of the requested program in table TLIEFTE8.
  - a. If the name is in the table, \$CALL attaches the program RSAPF=YES. If the program is from an authorized library and is linked AC=1, it will run authorized.
  - b. If the name is not in the table, \$CALL attaches the program RSAPF=NO, and it runs unauthorized.

#### Rules

1. Table IKJEFTE2 (linked with the TMP) must contain '\$', but not '\$X'.
2. \$ must be linked AC=1 in an authorized library. It is the only TL/1 module which must be linked AC=1.



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.6	Page	1.1
Section Name	TSO UTILITIES				
Subsection Name	Special Considerations				
Subject					

Date Issued 1-25-78

Replaces Issue

3. All commands invoked with '\$' entry point must be in authorized libraries; if a command from an authorized library is invoked with '\$', \$ will abend with a 306.

Note: TL/1 commands are currently in either SYS1.CMDLIB or SYS1.LPALIB, so this shouldn't be a problem.

4. Because TL/1 commands execute unauthorized when invoked with '\$X', they cannot perform restricted operations (e.g., issue restricted SVC's).

Example: Don't try testing \$DEGAS (which uses IEBCOPY) by saying '\$X DEGAS...'

5. Table TLIEFTE8 (used by \$CALL) should contain all programs which request restricted services and can be invoked with \$CALL (e.g., IEBCOPY).

These programs must be APF-authorized (i.e., linked AC=1) and must reside in authorized libraries.



Standard Oil  
Company (Indiana)

Manual	FSD-2	Section	24.7	Page	1
Section Name		TSO UTILITIES			
Subsection Name		Messages			
Subject					

Date Issued 01-18-78

Replaces Issue

Refer to FSD-3 (Operator's Guide) for full explanation of the following messages:

TL1001 OPEN FAILED FOR TL/1 COMMAND LIBRARY  
TL1002 COMMAND NAME MISSING OR INVALID  
TL1003 COMMAND NAMED IS NOT IN LIBRARY  
TL1004 LIBRARY ACCESS NOT AUTHORIZED FOR YOUR USE

TLI0806C, "NAME" NOT FOUND  
TLI0806P



Standard Oil  
Company (Indiana)

Manual FSD-2

Section 24.8

Page  
1

Date Issued 01-18-78

Section Name TSO UTILITIES

Subsection Name Debugging Aids

Replaces Issue

Subject

The following facilities should be useful in debugging TL/I commands:

Autolist Support  
Facility

Use 'LIST' or 'LIST(ON)' in command to get conditional  
debugging statements displayed at terminal.

\$X entry

Use '\$X' entry point to Command Processor '\$', which  
allows user to test TL/I command from a user load library.

TSO Test

Useful for debugging TL/I commands interactively.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual	FSD-2	Section	24.9	Page	1
Section Name	TSO UTILITIES				
Subsection Name	Programming Notes				
Subject					

For documentation of TL/I language itself, see FSD-4 Section 97:

TL/I Language Reference Manual  
TL/I Programmer's Guide  
TL/I Program Logic and Data Structures



Standard Oil  
Company (Indiana)

Manual

FSD-2

Section

24.10

Page  
1

Date Issued 01-18-78

Section Name TSO UTILITIES

Subsection Name History of Component Changes

Replaces Issue

Subject

Installation: January, 1978





Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-2

Section 24.11

Page  
1

Section Name TSO UTILITIES

Subsection Name History of Modification Installations

Subject

NONE

## MVS OPERATING SYSTEM

## Messages

## T

<u>Message</u>	<u>Explanation</u>
TLI001	OPEN FAILED FOR TL/1 COMMAND LIBRARY
Explanation:	The User has invoked the TL/1 Command system with the '\$X' or '@' entry point; the correct DDNAME (TLILIB or TLILIBX) is not currently allocated.
Response:	Allocate the correct load library(s) to DDNAME TLILIBX (for \$X entry) TLILIB (for @ entry)
TLI002	COMMAND NAME MISSING OR INVALID
Explanation:	TL/1 command module names must be eight characters or less and begin with a dollar sign. The name must be <u>specified without</u> the leading dollar sign when entering the command.
Response:	Enter correct command syntax
Example:	'\$ØPRINT' invokes the load module whose name is '\$PRINT'.
TLI003	COMMAND NAMED IS NOT IN LIBRARY ("name")
Explanation:	The TL/1 load module name (supplied in parentheses in message) was not found via BLDL. Either the name was misspelled or an incorrect load library specification was made.
Response:	Enter command with correct command name.
TLI004	LIBRARY ACCESS NOT AUTHORIZED FOR YOUR USE
Explanation:	The "@" entry point, supported for special services and debugging, will not function for users without OPER authority. TL/1 library support for normal users is via the "\$X" entry point.
Response:	none

## MVS OPERATING SYSTEM

## Messages

T

TLI0806C  
TLI0806P

"name" NOT FOUND

Explanation: This message is issued when the TLICALL/TLICP service routines fail to find the program or command processor specified in the service request. "Name" in the message indicates the missing program module.

Note: The last character of the message ID is 'P' for a called program or 'C' for a command processor.

Response: The user is running with:

- a) an invalid STEPLIB statement
- b) a system with one or more Corporate Utilities missing from U.ULUTIL
- c) command processors missing from the linklist.

Occurrences of this message should be reported to Center Support for commands other than \$CALL.

TLI013C  
TLI013P

"member" NOT FOUND

Explanation: This message is issued when the TLICALL/TLICP service routines fail to find the program or command processor in the tasklib specified in the service request. "Member" in the message indicates the missing member.

Note: The last character of the message ID is 'P' for a called program or 'C' for a command processor.

← Response: Correct FILE operand to specify library in which copy of program is to be found.

Occurrences of this message should be reported to Center Support for commands other than \$CALL.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 24.10

Page  
1

Date Issued 01-18-78

Section Name TSO UTILITIES

Subsection Name \$

Replaces Issue

Subject Component Description

#### Overview

Command Processor \$ is the basic interface between TL/I commands and the system. All '\$' commands are initially processed by this module, which then passes control to the appropriate TL/I command module.

\$ has 3 entry points: \$, @, \$X

#### Algorithm

-

#### Interfaces

None

#### Data Elements

DCB for TLILIB or TLILIBX

#### File Definitions

##### TLILIB

Used as STEPLIB if entry to command processor was '@'.

##### TLILIBX

Used as STEPLIB if entry to command processor was '\$X'.

#### Abstract

\$ searches for the appropriate TL/I module and loads it. After building and/or copying control blocks, \$ passes control to the TL/I module via ATTACH.

#### Cross References

See FSD-4 Section 97 for documentation of TL/I language.

#### Debugging Aids

The following facilities should be useful in debugging TL/I commands:

##### Autolist Support Facility

Use 'LIST' or 'LIST(ON)' in command to get conditional debugging statements displayed at terminal.

##### \$X Entry

Use '\$X' entry point to Command Processor '\$', which allows user to test TL/I command from a user load library.

##### TSO Test

Useful for debugging TL/I commands interactively.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 24.2.0

Page  
1

Date Issued 01-18-78

Section Name TSO Utilities

Subsection Name \$CALL

Replaces Issue

Subject Component Description

#### Overview

\$CALL is a TSO command which calls and executes a named program. If file (FI) or dataset (DA) operands are specified in the command, the files named will be used as TASKLIB by \$CALL. \$CALL is written in TL/I and not supported in batch.

#### Algorithm

None.

#### Interfaces

The named program is attached by the \$CALL macro.

#### Data Elements

None.

#### Abstract

\$CALL does some editing of command operands; if parameter 'DA' was specified, \$DDN is used to create a ddname and the file is allocated.

The batch bit (TLIBIT4) is checked; if on, an error message is sent to the terminal.

The \$CALL macro is issued, passing parameters from the command. If keyword 'UTIL' was specified in the command, a list of IKJ prefix filenames is passed (see IJK\$DDN in TLICOPY) as a substitute ddname list for a utility program.

\$CALL will execute the program specified by the 'MEMBER' parameter after searching:

- 1) TASKLIB specified in command.
- 2) System link list.

Return code set by \$CALL is checked and processed.

#### File Definitions

None.

#### Cross References

See FSD-4 Section 97 for documentation of TL/I language.

#### Debugging Aids

The following facilities should be useful in debugging TL/I commands:

##### Autolist Support Facility

Use 'LIST' or 'LIST (ON)' in command to get conditional debugging statements displayed at terminal.

##### \$X entry

Use '\$X' entry point to Command Processor '\$', which allows user to test TL/I command from a user load library.

##### TSO test

Useful for debugging TL/I commands interactively.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual

FSD-4

Section 24.3.0

Page  
1

Section Name TSO UTILITIES

Subsection Name \$COPY

Subject Component Description

#### Overview

\$COPY is a TSO command which will copy:

- a. Up to 16 members (or all members) from one PDS to another.
- b. One sequential dataset to another.
- c. A sequential dataset to a member of a PDS.
- d. A PDS member to a sequential dataset.

\$COPY has a separate alias and entry point \$DEGAS, which will compress up to 16 PDS's in place. The PDS(s) will be copied to a backup dataset before the compress depending on operands supplied in the command.

#### Algorithm

None.

#### Interfaces

None.

#### Data Elements

None.

#### File Descriptions

For \$DEGAS:

\$DINPUT dsname supplied in command, naming file to be compressed.

\$DBACKUP copy of dataset made before compress. Allocated by \$DEGAS as NEW CYL SPACE (15 10).

For \$COPY:

\$COPYIN dsname or ddname supplied in command as INDSN or INFI, naming file to be copied.

\$COPYOUT dsname or ddname supplied in command as OUTDSN or OUTFI, naming file to be copied to.

For \$COPY and \$DEGAS:

CTL (dsnames \$COPY.DATA, \$COPY.BACK, \$COPY.INPUT) control cards for IEBCOPY for compress, backup, and copy.

Control card file allocated as:

BLOCK(800) SPACE(10 35) NEW  
RECFM(F,B) LRECL(80) BLKSIZE(800)



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-4

Section 24.3.0

Page  
2

Section Name TSO UTILITIES

Subsection Name \$COPY

Subject Component Description

SYSPRINT output from IEBCOPY. Allocated as:

BLOCK(3509) SPACE(19 19) NEW  
TRACKS DS(PR)

where PR defaults to \$COPY.LIST, but may be set to '\*' by  
command operand PR.

#### Cross References

See FSD-4 Section 97 for documentation of TL/1 language.

#### Abstracts

##### \$DEGAS

Allocate SYSPRINT, either to terminal or dataset \$DEGAS.LIST.  
Prepare IEBCOPY control card for compress, and if 'RECOVER'  
specified, control card for backup. Then for each dataset  
specified in the command (operand INDSN), do the following:

1. Allocate a sequential backup file.
2. Call IEBCOPY to back up the PDS.
3. Report success or failure of backup operation to  
terminal user.
4. Call IEBCOPY to compress requested dataset in place.
5. Report success or failure of compress to terminal user.

##### \$COPY

Check command syntax and prompt for required parameters if  
not supplied (INDSN/INFI, OUTDSN/OUTFI). Allocate SYSPRINT  
either to terminal or \$COPY.LIST.

If input and output is PDS and no member(s) specified in  
INDSN,OUTDSN:

Prompt for memberlist (unless 'ALL' specified).  
If members being renamed, build array of new names.  
Build 'COPY' control card.  
If 'NOALL' specified, build 'SELECT' control card for  
each member.  
Call IEBCOPY.  
Free files \$COPYIN, \$COPYOUT, CTL, SYSPRINT.



Standard Oil  
Company (Indiana)

Date Issued

01-18-78

Replaces Issue

Manual

FSD-4

Section 24.3.0

Page  
3

Section Name

TSO UTILITIES

Subsection Name

\$COPY

Subject

Component Description

If either input or output is not PDS, or member specified in INDSN, OUTDSN:

allocate IKJIN,IKJPRINT,IKJUT1,IKJUT2.  
call IEBGENER to do copy.  
free IKJIN,IKJPRINT,IKJUT1,IKJUT2.

#### Debugging Aids

The following facilities should be useful in debugging TL/I commands.

Autolist  
Support  
Facility

Use 'LIST' or 'LIST(ON)' in command to get conditional debugging statements displayed at terminal.

\$X entry

Use '\$X' entry point to command processor '\$', which allows user to test TL/I command from a user load library.

TSO test

Useful for debugging TL/I commands interactively.





Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual	FSD-4	Section	24.4 .0	Page	1
Section Name		TSO UTILITIES			
Subsection Name		\$PDS			
Subject		Component Description			

#### Overview

\$PDS is a TSO command which allows a TSO user to access and manipulate the directory and selected members of a PDS.  
(\$PDS is a TL/I driver for routine TLIPDQ (BAL) which actually processes the command.)

#### Algorithm

None

#### Interfaces

None

#### Data Elements

The following control blocks are read by \$PDS:

##### DCB

(Data Control Block) Data pertinent to current use of a dataset

##### DEB

(Data Extent Block) Extension for information in the DCB: used by \$PDS to get number of extents for dataset.

##### DSCB

(Data Set Control Block) Describes characteristics and up to 3 extents of a dataset; used by \$PDS to get volser.

##### UCB

(Control Block) Describes the characteristics of a device to the I/O Supervisor; used by \$PDS to get last TTR used in dataset.

#### File Definitions

None

#### Abstracts

##### \$PDS

\$PDS is a TL/I driver for BAL routine TLIPDQ; it calls TLIPDQ and then issues \$QUIT.

##### TLIPDQ

Module TLIPDQ establishes work areas and pointers, and calls IKJSCAN to get the command name and IKJPARS to get command operands from the input buffer. The PDS specified in the command is allocated, and control is passed to the appropriate subcommand processing routine.

Module TLIPDQ handles ATTN processing.

##### ALIAS

Subcommand processing routine ALIAS issues a STOW to add an alias for a PDS member. If the PDS is a load library, APF information is checked first, and SSI data may be modified.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.1

Page  
7

Section Name TSO UTILITIES

Date Issued 1-20-78

Subsection Name TL/1 Language Reference Manual

Replaces Issue

Subject Parsing the Command Buffer

Here is the first part of the \$ALLOC command. This demonstrates the code necessary to generate the above syntax:

```
ALLOC: /* SAMPLE COMMAND OF TL/I SYNTAX */
      % Include (TLICOPY): /* CONVERT PL/I to TL/I*/
/*
```

GENERATE THE \$ ALLØC SYNTAX

```
*/
SDCL(FILE,8);          /* DEFINE                */
SDCLIST(DATASET,56);   /* ALL                */
SDCL(DISP,3,'SHR');     /* THE                */
SDCL(CØMMENT, 128, 'DØN'T DELETE'); /* KEYWORDS          */
$ABBREV((FILE, FI), (DATASET,DA),(CØMMENT,CMT));
$QUOTED(DATASET);
$ASIS(CØMMENT):
```

\$PARSE; /\* VALUES FROM COMMAND BUFFER ARE NOW ASSIGNED. \*/

```
$PØMPT(FILE); /* MUST HAVE FILE */
$PØMPT(DATASET(1)); /* AT LEAST FIRST DATASET */
$PØMPT(DISP,'ØLD','MØD','SHR') /* AND VALID DISP */
```

### Defining Action Keywords:

Defining each keyword.

\$KEY (name);

name: The name of the keyword. It can be up to 28 characters long.

Initializing all action keywords.

\$KEYDEF;

The \$KEYDEF statement must follow all \$KEY statements; this statement initiates processing to remove action keywords from the command buffer and initialize the associated flags.

### Defining Context-Sensitive Keywords:

Defining a scalar keyword.

\$DCL(name,length[, 'value']);

name: The name of the keyword. It can be up to 28 characters long.

length: The maximum length the value assigned to the keyword can be.

value: A default value the keyword is to have if not found in the buffer.



Standard Oil  
Company (Indiana)

Manual	FSD-4	Section	97.1	Page	6
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Language Reference Manual				
Subject	Parsing the Command Buffer				

Example:

This example is used to demonstrate how a specific TL/I command is documented, entered, and coded. Here it is documented (SYNTAX):

```
$ALLØC FILE:FI(8) DATASET#:DA/QUØTED(56)
      DISP(3;'ØLD','MØD','SHR')
      CØMMENT:CMT/ASIS(128)
```

```
REQUIRED - FILE, DATASET, DISP
DEFAULT  - DISP(SHR), CMT(DØN'T DELETE)
```

What is meant by the above:

FILE is the first keyword. It was the abbreviation FI and can be eight characters long.

DATASET is the second keyword. It has the abbreviation DA and the '#' implies that it can be a list of values. Each value can be 56 characters and if quoted, the quotes will be left.

DISP is the third keyword. It has no abbreviation and can be three characters long. It has three legal values. It is assigned a value other than these three, a valid reply will be prompted.

COMMENT is the fourth keyword. It has the abbreviation CMT and can be up to 128 characters long. Its value is not to be converted to upper case.

Here is an example of the use of this command:

```
$ alloc syslib ('cap.maclib' 'cap.reclib' 'cap.reclib2'
  programs.data 'sysl.maclib') , , 'This is an example'
```

Here are other ways of entering the same thing:

```
$ alloc fi(syslib da('cap.maclib' 'cap.reclib' 'cap.reclib2'
  programs.data 'sysl.maclib') Cmt(This is an example)
```

```
$ alloc da(replaced) cmt(This is an example) syslib shr da#1('cap.maclib'
  'cap.reclib' 'cap.reclib2' programs.data 'sysl.maclib')
```

```
$ alloc cmt(This is an example) syslib ('cap.maclib'
  'cap.reclib' 'cap.reclib2' programs.data 'syslib.maclib')
```

```
$ alloc da( , , , programs.data) da('sysl.maclib') syslib
  da#2('cap.reclib') da('cap.reclib2') shr
  da#1('cap.maclib') (This is an example)
```



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
5

Section Name: TSO UTILITIES

Subsection Name: TL/I Language Reference Manual

Subject Parsing the Command Buffer


- 11) A keyword list assumes that the first element of the list in the buffer will be assigned to the first element of the list in the command. This can be overridden by specifying the number to be assigned the first element in the list as follows: KEYWORD#N(LIST). N is a one or two-digit number and defines the logical beginning entry in the list (see example below).
- 12) Keywords may be defined with these additional attributes (besides that of being a list or not):
  - A) ASIS: the value of the keyword is not converted to upper case.
  - B) QUOTED: if the string is entered as a positional quoted operand, the outside quotes are not removed (e.g., data set names).
  - C) ABREV: an abbreviation may be assigned to each keyword.
  - D) DEFAULTS: keywords may have a default.
  - E) REQUIRED: a keyword may require a value.
  - F) VALID: a keyword may have a required and specific set of values.
- 13) If the buffer ends with an unbalanced quote, a quote is added. If it ends with one or more missing right parenthesis, they are added.
- 14) TL/I syntax solution and command flow is turned on if list (ON) is found in the command buffer. (List (OFF) in the buffer is a NO-OP.)

#### Action Keywords:

Action keywords are recognized wherever they occur within the command (following the command name). They are recognized by the command parsing routines and are removed from the command buffer before the processing of context-sensitive keywords.  
Example:

CALL COPYJOB UTIL

'UTIL' is an action keyword (see help entry for \$CALL), and is used by the \$CALL command processor to select a list of utility ddnames to be used by the program 'COPYJOB'.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page	4
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Language Reference Manual				
	Subject	Parsing the Command Buffer				
Date Issued	1-20-78					
Replaces Issue						

TL/I supports two types of operands:

Context-Sensitive Keywords  
Action Keywords

#### Context-Sensitive Keywords:

Unlike standard TSO operands, which are keywords only or positional only, quoted string only or parenthesized string only; depending on the context under which an operand is entered, a context-sensitive keyword is considered as a positional operand, a keyword operand, a quoted string operand, or a parenthesized string operand.

Here are the rules for TL/I's context-sensitive keywords.

- 1) Each operand is defined with a default position in the input buffer: first position, second position, etc.
- 2) Each operand also is defined with a keyword name (e.g., file, da).
- 3) Operands may be separated by either one or more blanks or a single comma, and zero or more blanks on either side of the comma.
- 4) If more than one comma is found with only intervening blanks, then a missing positional operand is implied (e.g., , , means one missing operand; , , , , means three missing operands).
- 5) The command buffer is logically scanned twice. In the first scan, operands entered as keywords are assigned and removed from the buffer.
- 6) In the second scan, only positional operands are assigned in order, to the unassigned context sensitive keywords in the command buffer, with Rule 4 above applying.
- 7) In the first scan all operands are handled as parenthesized strings.
- 8) In the second scan, as each operand is scanned, the first left parenthesis or quote mark determines whether the string is a parenthesized or quoted string (e.g., quoted strings: HI'THERE'GUY,'DA(MEM)', 'P(','DON'T'; paren strings: HI(THERE)GUY,(DA(MEM)),(P'),(DON'T)).
- 9) Some commands have operands which are lists of values. The list itself must be a parenthesized string, but the individual items in the list may be either paren or quoted strings (e.g., DA('HI('.,HELLO,TO(MEM),'DON'T')).
- 10) If a keyword is found more than once in the buffer, and the keyword can refer to a list of values, the lists are concatenated together in order (see example below).



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
3

Section Name TSO UTILITIES

Subsection Name: TL/1 Language Reference Manual

Subject WHY TL/1

### A Time Sharing Control Language (TL/1)

The current command language facilities of TSO do not provide adequate flexibility for controlling user interaction throughout a session.

The current command language has too few primitives, and lacks logical control functions, branching capabilities, and terminal interaction from CLIST facilities.

The current command language is at a very low level, the primitives in its set interacting with the first level routines of the Time-Sharing software. While this level of access is very necessary for the systems programmer, the users of the Time-Sharing system also write command procedures, and the overhead in training many users to an adequate level of competence in the current CLIST facility is too high in overhead to justify. Clearly, what is needed is a HIGH LEVEL command language for TSO.

While the EXEC language of VM/370-CMS answers some of the shortcomings in the TSO command language set, the CMS EXEC language is still a low level control language, and the very high flexibility is paid for with very high user knowledge requirements. The CMS EXEC Language then can be considered an acceptable command language for the use of Time-Sharing system maintenance and user personnel.

This language uses the existing TSO command language (with some CMS-like additions) as the primitive set for a higher level language which is noninterpretative, user-oriented, and easily interfaced.

Also, as IBM has unified the interfaces to the operating system from both the batch and foreground, the number of "control" languages which a user must know can be reduced, hopefully. Now the user must know a batch "control language": JCL and a foreground "control language": CLISTS; neither of which fully qualifies as a language, much to the frustration of their users. Now it is potentially possible, as of VS2R2, to have a single HIGH LEVEL, low frustration level, and powerful control language for OS.

Although the current implementation accesses the primitive set by invoking an assembly language interface to pass a TSO command image string to the TSO TMP or execute a program, a more formal implementation of the language which produced compile-time direct interfaces would be much more effective. The authors of this implementation have taken the position that modification of the PL/1 compiler is too complex to be attempted in the initial implementation.

The current PL/1 Optimizing Compiler is a reasonably efficient generator of code, and does represent a usable implementation of a high-level language with a syntax, which is very well suited to the generation of command and control programs.

The current TL/1 implementation uses as a base the PL/1 language. This language, together with a specialized library of callable subprograms, provides an adequately flexible base for the examination of the feasibility of a high-level Time-Sharing and Batch Control Language.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
2

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject Introduction

This document, and the language it describes, are still in a state of development. This document and IBM's PL/1 Optimizing Compiler Language Reference Manual (SC33-0009), describes the Language TL/1.

The authors will appreciate any comments or corrections. Correspondence should be sent to the addresses shown below.

#### Correspondence

Standard Oil Company (Indiana)  
Software Development Division, Mail Code 0505  
200 East Randolph Drive  
Chicago, Illinois 60601



Standard Oil  
Company (Indiana)

Date issued 1-20-78

Replaces issue

Manual FSD-4

Section 97.1

Page  
1

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject Table of Contents

## INTRODUCTION

### WHY TL/1

### PARSING THE COMMAND BUFFER

#### GENERAL DISCUSSION

#### EXAMPLE

#### DEFINING ACTION KEYWORDS

#### DEFINING CONTEXT-SENSITIVE KEYWORDS

#### DESCRIBING KEYWORDS

#### PARSING

### TL/1 STATEMENTS

### TL/1 FUNCTIONS

### ON CONDITIONS

### INCLUDE GROUPS





Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual

FSD-4

Section 24.13 .0

Page  
1

Section Name TSO Utilities

Subsection Name \$CATFI

Subject Component Description

Overview	This command concatenates a list of filenames, each of which has been previously allocated.
Algorithm	A list of DD names is moved to an array, one DD-name to each element, up to a maximum of 15. The array elements are used as arguments for the \$CAT statement.
Interfaces	None.
Data Elements	None.
File Definition	None.
Abstracts	None.
Cross References	There are no modifications to Type 1 software for this component.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual

FSD-4

Section

24,12.0

Page

1

Section Name

TSO Utilities

Subsection Name

\$DDALLOC

Subject

Component Description

#### Overview

This command takes as input a complete DD-statement (including continuation cards) and generates an ALLOCATE command.

#### Algorithm:

JCL entered at the terminal is concatenated in a work area. The DD card parameters are located and moved to separate fields where they form part of an ALLOC command. The \$CMD statement is used to issue the command.

#### Interfaces

None.

#### Data Elements

None.

#### File Definition

None.

#### Abstracts

None.

#### Cross References

There are no modifications to Type 1 software for this component.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual

FD-4

Section

24.11.0

Page

1

Section Name

T30 Utilities

Subsection Name

\$DBFREE

Subject

Component Description

#### Overview

\$DBFREE is a tso command which provides a way for users to de-allocate databases from the T30 foreground.

#### Algorithm

The name of the PSB or DBN card, supplied by the user, is used to search the 'ARISPSDS' data set for the specified PSB or DBN file. If a PSB card has been entered, the DBN cards contained in the file are read and are, in turn, used to locate the DBN files specified. If the DBN file contains more DBN cards, these are processed as before, until ultimately, a DBN file is located containing DB cards for the database datasets to be freed. The DB card records are extracted and a FREE command issued for each.

#### Interfaces

None.

#### Data Elements

None.

#### File Definition

ARISPSDS - partitioned data set (fixed blocked, LRECL=30) containing PSB and/or DBN file(s).

#### Abstracts

##### GETPSB

a) this module is loaded at execution time

b) GETPSB locates a specified PSB file in the ARISPSDS data set and then reads the file, passing the records back to the calling program, one at a time, along with a return code.

##### GETDBN

a) this module is loaded at execution time

b) GETDBN searches the ARISPSDS directory until it locates the specified DBN file. It then reads the file and passes the records, one at a time, back to the calling program along with a return code.

#### Cross References

There are no modifications to Type 1 software for this component.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual	FSD-4
Section Name	TSO Utilities
Subsection Name	\$DBALLOC
Subject	Component Description

Section 24.10 .0

Page  
1

#### Overview

\$DBALLOC is a tso command which provides a way for users to allocate databases from the TSO foreground.

#### Algorithm

The name of the PSB or DBN card, supplied by the user, is used to search the 'ARISPDOS' data set for the specified PSB or DBN file. If a PSB card has been entered, the DBN cards contained in the file are read and are, in turn, used to locate the DBN files specified. If the DBN file contains more DBN cards, these are processed as before, until ultimately, a DBN file is located containing DD cards for the database datasets to be allocated. The DD card records are extracted and an ALLOC command issued for each.

#### Interfaces

None.

#### Data Elements

None.

#### File Definition

ARISPDOS - partitioned data set (fixed blocked, LRECL=80) containing PSB and/or DBN file(s).

#### Abstracts

##### GETPSB

- a) this module is loaded at execution time
- b) GETPSB locates a specified PSB file in the ARISPDOS data set and then reads the file, passing the records back to the calling program, one at a time, along with a return code.

##### GETDBN

- a) this module is loaded at execution time
- b) GETDBN searches the ARISPDOS directory until it locates the specified DBN file. It then reads the file and passes the records, one at a time, back to the calling program along with a return code.

\$DBALLOC - this is a tso utility executed by \$DBALLOC which builds and issues an ALLOCATE statement for each card passed to it.

\$CATF1 - this is a tso utility executed by \$DBALLOC which concatenates a list of file names.

#### Comments

There are no modifications to Type 1 software for this component.



Standard Oil  
Company (Indiana)

Date Issued

01-18-73

Replaces Issue

Manual

FSD-4

Section 24.9 .0

Page  
1

Section Name

TSO UTILITIES

Subsection Name

\$WHO

Subject

Component Description

## Overview

\$WHO is a TSO command used to display all jobs or users enqueued on a particular dataset.  
(\$WHO written in TL/I; calls TLIENQ# and TLIENQ2# (ASM))

## Algorithm

-

## Interfaces

None.

## Data Elements

The following control blocks/data areas are read by \$WHO:

### QCB

Queue Control Block - identifies a request for a resource being serialized by ENQ/DEQ/RESERVE.  
Major QCB - corresponds to name on ENQ.  
Minor QCB - corresponds to minor name on ENQ.

### QEL

Queue Element - identifies task that issued ENQ.

### CSCB

Command Schedule Control Block - created by LOGON for life of TSO terminal session; contains run time, job description passed to command except routines from command scheduling routines.

## File Definition

None.

## Cross References

See FSD-4 Section 97 for documentation of TL/I language.

## Abstract

\$WHO  
(mainline)

The mainline of \$WHO edits the dsname supplied in the command and passes this name, along with major queue name 'SYSDSN' to subroutine TLIENQ#.

### TLIENQ#

This routine extracts major and minor queue names from the parameter passed to it by the calling routine, then retrieves major and minor QCB which matches these names. A table is constructed with information from the QCB's and related QEL's and CESD's. Routine TLIENQ2# is then called to do I/O.

### TLIENQ2#


This routine formats the enqueue information retrieved by TLIENQ#, does a FPUT and frees all work areas.

## Debugging Aids

The following facilities should be useful in debugging TL/I commands:

### Autolist Support Facility

Use 'LIST' or 'LIST (ON)' in command to get conditional debugging statements displayed at terminal.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section 24.8.0	Page 1
	Section Name	TSO UTILITIES		
	Subsection Name	\$TR		
	Subject	Component Description		
Date Issued	01-18-78			
Replaces Issue				

Overview	\$TR is a TSO command which allows the user to specify print, punch, and copy transmissions to designated locations and specify offline process information for TAPE, SCHED, FICHE, and PLOT (written in TL/I; calls STSALI).
Algorithm	None.
Interfaces	STSALI
Data Elements	None.
File Definitions	None.
Abstract	<p>\$TR issues a TSO 'PROF NOPROMPT', allocates files TLIPRINT and SYSLIB, and builds a TR request from parameters supplied in the command.</p> <p>\$TR does a FETCH for STSALI, calls STSALI (passing a null parameter, return code, the edited input string, and a ddname string) and then RELEASES STSALI.</p>
Cross References	See FSD-4 Section 97 for documentation of TL/I language.
Debugging Aids	The following facilities should be useful in debugging TL/I commands:
Autolist Support Facility	Use 'LIST' or 'LIST (ON)' in command to get conditional debugging statements displayed at terminal.
\$X entry	Use '\$X' entry point to Command Processor '\$', which allows user to test TL/I command from a user load library.
TSO test	Useful for debugging TL/I commands interactively.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-4

Section 24.7.0

Page  
2

Section Name TSO UTILITIES

Subsection Name \$RESET

Subject Component Description

## Debugging Aids

The following facilities should be useful in debugging TL/I commands:

Autolist Support Facility--Use 'LIST' or 'LIST(ON)' in command to get conditional debugging statements displayed at terminal.

\$X Entry--Use '\$X' entry point to command Processor '\$', which allows user to test TL/I command from a user load library.

TSO Test--Useful for debugging TL/I commands interactively.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-4

Section 24.7.0

Page  
1

Section Name TSO UTILITIES

Subsection Name \$RESET

Subject Component Description

#### Overview

\$RESET is a TSO command which will free files specified in command parameters, or free specified files and reallocate them to the terminal.

#### Algorithm

None

#### Interfaces

None

#### Data Elements

\$RESET reads the TSO user's DSAB chain, which describes the status of all files allocated by that TSO user. The DDNAME for each file is obtained from the TIOT DD entry for that file (pointed to by the DSAB), and the DSNAME is obtained from the JFCB pointed to by the TIOT entry.

#### Field Descriptions

None

#### Cross References


See FSD-4 Section 97 for documentation of TL/1 language.

#### Abstracts

The TSO user's DSAB chain is located by referencing TCB, JSCB, and QDB. DSAB's which describe files allocated by the TSO user are then processed as follows:

- a. TIOEDDNM (in the TIOT and DD entries pointed to by the DSAB) is compared with any file names specified in the command (parameter FI). If a match is found, a TSO FREE command is issued for that file.
- b. Prefixes of TIOEDDNM are compared with command operand TYPE. If a match is found, a TSO FREE command is issued for the file.
- c. TIOEDDNM is compared with command operand TERM. If a match is found, a TSO FREE command is issued for the file; after the DSAB chain is completely read, all files specified by 'TERM' are reallocated to the terminal.
- d. The DSNAME in the JFCB is compared with command operand DA. If a match is found, a TSO FREE command is issued for the dataset, unless the dataset is a member of a concatenation.
- e. If keyword 'ALL' is specified in the command, a FREE ALL command is issued.
- f. If keyword 'SYSOUT' is specified in the command, all SYSOUT files are freed.



 Standard Oil Company (Indiana)	Manual	FSD-4	Section	24.6.0	Page	2
	Section Name	TSO Utilities				
	Subsection Name	\$PRINT				
	Subject	Component Description				
Date Issued	01-18-78					
Replaces Issue						

SYSOUT (class) - where 'class' is as specified in command or defaults to 'S'.

DCB parameters as defined above.

#### Cross References

See FSD-4 Section 97 for documentation of TL/I language.

#### Abstract

If command is \$PUNCH, CARD and FORM parameters are set; processing continues under \$PRINT.

If command is \$SEND, a \$TR request is built from parameters supplied in command for up to 16 datasets or members of a specified PDS.

If command is \$PRINT, processing depends on value for TYPE (default is SYSOUT).

If TYPE = STS:

Each dataset (up to 16) or PDS member (up to 16) is copied (using TLICOPY or IEBPTPCH, respectively) and a \$TR request is built using parameters in command.

If TYPE = SYSOUT:

Each dataset, PDS, or PDS member is copied to a SYSOUT file, using TLICOPY for non-PDS datasets and IEBPTPCH for entire PDS or specific PDS members.

#### Debugging Aids

The following facilities should be useful in debugging TL/I commands:

##### Autolist Support Facility


Use 'LIST' or 'LIST (ON)' in command to get conditional debugging statements displayed at terminal.

##### \$X entry

Use '\$X' entry point to command Processor '\$', which allows user to test TL/I command from a user load library.

##### TSO test

Useful for debugging TL/I commands interactively.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	24.6 .0	Page	1
	Section Name	TSO Utilities				
	Subsection Name	SPRINT				
	Subject	Component Description				
Date issued	01-18-78					
Replaces issue						

**Overview** \$PRINT is a TSO command which will (depending on alias and value of TYPE parameter specified in command) either:

- 1) print a requested list of datasets, PDS's or PDS members on any printer).
- 2) punch a requested list of datasets, PDS's or PDS members.
- 3) issue a \$TR request to print requested datasets or PDS members.

(Written in TL/I; calls IEBPTPCH and TLICOPY.)

**Algorithm** -

**Interfaces** None.

**Data Elements** None.

**File Definition**

\$PRINT

If Type = STS is specified in a \$PRINT command, the requested dataset is copied to a file with the following characteristics:

```

FI ($PRINT)
DA (STSDSN) - constructed from TLIUSER, $ITNAME, $FCC

NEW
TRACKS SPACE (#) where '#' is specified in command or
defaults to 38

```

If the file is being used to copy PDS members (IEBPTPCH) DCB parameters are:

```

RECFM (F, B, A)  LRECL (133)  BLKSIZE (3458)

```

If the file is being used to copy non-PDS datasets (TLICOPY) DCB parameters are:

```

RECFM (U, B, A)  LRECL (137)  BLKSIZE (6136)

```

If TYPE = SYSOUT is specified in a \$PRINT command, the requested dataset is copied to a SYSOUT file with the following characteristics:



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-4

Section 24.5.0

Page  
1

Section Name TSO UTILITIES

Subsection Name \$PMAP

Subject Component Description

## Overview

\$PMAP is a TSO command used to display attribute information associated with a dataset (or list of up to sixteen datasets).

\$PMAP is written in TL/I; calls BAL subroutine TLIDSX to retrieve and format attribute information.

## Algorithm

-

## Interfaces

None.

## Data Elements

\$PMAP reads the following control blocks:

### DSCBI

Identifier Dataset Control Block-describes the characteristics and up to three extents of a dataset on a direct access volume.

## File Definitions

None.

## Abstract

\$PMAP displays a header line at the terminal, edits the dsname from the command, and calls subroutine TLIDSX to retrieve information about the dataset. \$PMAP then displays the data returned by TLIDSX.

### TLIDSX

This module reads the format, DSCB for the dataset, formats the required data, and passes this formatted data and the DSCB back to the \$PMAP main routine.

## Cross References

See FSD-4 Section 97 for documentation of TL/I language.

## Debugging Aids

The following facilities should be useful in debugging TL/I commands:

### Autolist Support Facility

Use 'LIST' or 'LIST (ON)' in command to get conditional debugging statements displayed at terminal.

### \$X entry

Use '\$X' entry point to Command Processor '\$', which allows user to test TL/I command from a user load library.

### TSO test

Useful for debugging TL/I commands interactively.



Standard Oil  
Company (Indiana)

Date Issued 01-18-78

Replaces Issue

Manual FSD-4

Section 24.4.0

Page


3

Section Name TSO UTILITIES

Subsection Name \$PDS

Subject Component Description

Cross References	See FSD-4 Section 97 for documentation of TL/I language.
Debugging Aids	The following facilities should be useful in debugging TL/I commands:
Autolist	Use 'LIST' or 'LIST(ON)' in command to get conditional
Support Facility	debugging statements displayed at terminal.
\$X Entry	Use '\$X' entry point to Command Processor '\$', which allows
	user to test TL/I command from a user load library.
TSO Test	Useful for debugging TL/I commands interactively.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	24.4.0	Page	2
	Section Name	TSO UTILITIES				
	Subsection Name	\$PDS				
	Subject	Component Description				
Date Issued	01-18-78					
Replaces Issue						

ATTR	Subcommand processing routine ATTR displays attributes of the named PDS; attributes may be changed before display. If the PDS is a loadlib, APF data may be changed. The module issues a STOW to change attributes as specified in the command, and then displays all attributes.
CHANGE	Subcommand processing routine CHANGE issues a FREEMAIN to free the current command buffer and returns control to TLIPDQ to process a new PDS.
DISPLAY	Subcommand processing routine DISPLAY displays each member name in the directory for the named PDS.
HELP	Subcommand processing routine HELP links to the TSO HELP command processor.
HISTORY	Subcommand processing routine HISTORY gets and formats history information for a load library. The module reads the directory with a BLDL, and does EXCP's to read ESD entries and IDR records.
LIST	Subcommand processing routine LIST locates the PDS member specified in the command (if the PDS is not a load library), does EXCP's to read the member, and displays the member.
MAP	Subcommand processing routine MAP locates the directory entry for a load library and formats and displays ESD information.
OPTIONS	Subcommand processing routine OPTIONS displays the list of available \$PDS operand options.
RENAME	Subcommand processing routine RENAME issues a STOW with change option to rename a PDS member as specified in the command.
SCRATCH	Subcommand processing routine SCRATCH issues a STOW with delete option to scratch the specified PDS member.
USAGE	Subcommand processing routine USAGE scans the directory entry for the dataset specified in the command and displays directory information. The module also reads information from the DEB, UCB, and DSCB, computes total space used in the dataset, and displays usage information.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual

FSD-4

Section

97.1

Page

8

Section Name

TSO UTILITIES

Subsection Name

TL/1 Language Reference Manual

Subject

Parsing the Command Buffer

Defining a list keyword.

```
$DCLIST(name,length[,('value1',...,'valuen')]);
```

name: The name of the keyword. It can be up to 28 characters long. It defines a vector sixteen entries long.

length: The maximum length the values assigned to each entry in the list may be.

value<sub>i</sub>: The default value of the *i*th member of the list. If only the value of the first entry is required, the inner parentheses are not required. If more than one value is entered, the intervening values should be specified as ",", etc.,

RULE 1: The order the \$DCL and \$DCLIST's are entered is the assumed order of the operands in the buffer.

RULE 2: These two statements generate the declarations of the variables: no other declarations are required.

RULE 3: A maximum of 32 operands is allowed.

Resetting the maximum key limit.

```
$NUMKEYS(number);
```

number: The maximum number of keys in the input buffer or any occurrence of \$NEXT.

RULE 1: The default (no \$NUMKEYS verb specified) is 32, which is large enough for most situations.


RULE 2: This verb must be entered before the \$PARSE verb.

Describing The Keywords:

```
$ABREV((name,abbrev)[,(name2,abbrev2)[,...]]);
```

name: The keyword name being given the abbreviation.

abbrev: The abbreviation that this keyword can be known as within the command buffer. NOTE: This abbreviation allows the command user to eliminate unnecessary typing, it is not a valid alias for 'name' when coding the command itself.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page	9
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Language Reference Manual				
	Subject	Parsing the Command Buffer				
Date Issued	1-20-78					
Replaces Issue						

\$ASIS(name[,name2[,name3[,...]]]);

name: The name of the keyword. The data entered for this keyword will not be converted to upper case. (caveat emptor)

\$QUOTED(name[,name2[,name3[,...]]]);

name: The name of the keyword. If the value of this keyword is entered as a positional quoted string, the quotes will not be removed.

RULE 1: No more than fifteen keyword names can be described in any one descriptive verb, however, as many descriptive verbs as necessary can be used to describe the keywords.

RULE 2: A name may not have more than one abbreviation.

#### Parsing The Keyword:

\$PARSE;

This verb actually accomplishes the parse. It has no operands and is entered as shown.

\$PROMPT(name[,value2[,...]]];

name: The name of the keyword. It may not refer to a list keyword name unless subscripted.

value: A legal value for the keyword. If not legal values are specified, then prompting is done if the keyword contains null. If one or more values is specified, then prompting is done if the keyword does not equal one of these values. No more than fourteen values may be specified, and if a value is not in quotes, then it must be the name of a character string.

\$NEXT(string);

string: The name of a string to be parsed. This statement reinitializes the parsing routines and must be followed by a full set of parsing statements uniquely describing the possible contents of string. Its primary use is for describing subfields in previously parsed keywords. In that case, string would be the name of the keyword.



Standard Oil  
Company (Indiana)

Date issued 1-20-78

Replaces issue

Manual FSD-4

Section 97.1

Page  
10

Section Name TSO UTILITIES


Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

In addition to standard PL/I statements, several TL/I statements exist which provide capabilities frequently of use in command processors. These statements, documented on the following pages, are:

\$ABPUT  
\$CALL  
\$CAT  
\$CMD  
\$FAIL  
\$GET  
\$LINK  
\$PUT  
\$QUIT  
\$RCPUT  
\$STAK  
\$STOR



 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page	11
	Section Name:	TSO UTILITIES				
	Date Issued	1-20-78	Subsection Name:	TL/1 Language Reference Manual		
	Replaces Issue		Subject	TL/I Statements		

\$ABUT: Issue Abend message.  
\$ABPUT;

RULE 1: ECTRTCD must contain the abend code and ECTSCMD must contain the offending program's or command's name.

RULE 2: TLIHIGH will be updated to contain the highest abend code or return code discovered.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
12

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

\$CALL: EXECUTE A PROGRAM:

\$CALL (member[,parmstr[,clock[,file[,ddlist[,pageno]]]]]);

member: The program to be executed. This is an expression which is converted to a CHAR (8) VARYING string.

parmstr: The parameter string passed to the program. This is an expression which is converted to a CHAR (256) VARYING string.

clock: The value of a 322 clock specified in .01 seconds. A value of 0 implies no 322 clock. This is an expression which is converted to FIXED BIN (31).

file: The name of the DD card which points to the library or libraries searched for the program. If missing or null, the STEPLIB-JOBLIB-LNKLIB libraries are searched. This is an expression which is converted to a CHAR (8) VARYING string.

ddlist: A string containing nine-character ddname entries.

pageno: A four-character page number.

RULE 1: ddlist and pageno are described more fully in the OS UTILITIES manual in the appendix concerning program invocation of the utilities. Most batch programs do not use these parameters, and therefore ignore them.

RETURNS: SIGNAL CONDITION (RC) if program returns nonzero return code. Return code placed in ECTRTCD.

SIGNAL CONDITION (ABEND) if program abends. Abend placed in ECTRTCD. IF ECTRTCD < 4096 then abend is user, otherwise it is a system abend of ECTRTCD/4096.

Examples:

\$CALL('IEHLIST');

\$CALL(PROG(I),PARM(I),((3\*60)+10)\*100);

\$CALL(PRE||'FIX',,, 'MYLIBS');

\$CALL('IEBGENER',,,, 'SYSIN SYSPRINT INPUT OUTPUT');

Restriction: Clock expiration does not produce a dump.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
13

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

\$CAT: Concatenation:

```
$CAT(file,file2[,file3[,file4,...,]]);
```

file: The filename to which the datasets, pointed to by file and file2 through file<sub>n</sub>, are to be concatenated. The concatenation is in the order of the filenames specified.

file<sub>i</sub>: The i<sup>th</sup> filename points to the dataset which will become the i<sup>th</sup> member of the concatenation.

RULE 1: No more than fifteen and no fewer than two filenames may be specified.

RULE 2: Each filename is a character expression which must resolve to a character string of eight or less characters.

RULE 3: \$CAT ignores all filenames from the first null (") filename onward.

RETURNS: Return code in TLICODE.  
8=only one filename.

Examples:

```
$CAT('SYSLIB','SYSLIB','SYS' || SUBSTR(TIME,4,5));
```

```
$CAT(FI(I),FI(I+1));
```



Standard Oil  
Company (Indiana)

Manual	FSD-4	Section	97.1	Page	14
Section Name		TSO UTILITIES			
Subsection Name		TL/1 Language Reference Manual			
Subject		TL/I Statements			

\$CMD: Execute TSO Command

```
$CMD(command[,clock[,file]]);
```

command: The TSO command to be executed. It is an expression which, when evaluated, results in a character string of not more than 256 characters.

clock: The value of a 322 clock specified in .01 seconds. A value of 0 implies no 322 clock. This is an expression which is converted to FIXED BIN(31).

file: The name of the DD card which points to the library or libraries searched for the command. If missing or null, the STEPLIB-JOBLIB-LINKLIB libraries are searched. This is an expression which is converted to a CHAR(8) VARYING string.

RULE 1: Clists must be executed explicitly.

Rule 2: If the command has subcommands, the subcommands will be obtained from the current input medium. Therefore, if the TL/I command was initiated from a clist, they would be picked up there; otherwise, the subcommands are picked up from the terminal.

RULE 3: If the command has subcommands and the TL/I program wishes to specify subcommands before the command goes to the current input medium, the \$STAK and \$STOR statements must be used prior to the \$CMD statement.

RULE 4: This statement changes the value of ECTRCDF (SYSRC). If the TL/I command wishes to preserve SYSRC, this field must be saved and restored.

RETURNS: SIGNAL CONDITION (RC) if command returns nonzero return code. Return code placed in ECTRTCD.

SIGNAL CONDITION (ABEND) if command abends. Abend placed in ECTRTCD. If ECTRTCD 4096 the abend is user, otherwise it is a system abend of ECTRTCD/4096.

Examples:

```
$CMD('ALLOC FI('||FI(I)||') DA('||DA(I)||') SHR');
```

```
$CMD('EDIT' || MYIN || 'DATA');
```

```
$CMD('MYCMD OPS',3*100,'MYCMDLIB');
```

Restriction: Clock expiration does not produce a dump.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
15

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

**\$FAIL:** Issue abnormal termination message to terminal.

**\$FAIL;**

No parameters.

**\$FAIL** will cause the following message to be displayed at the user's terminal:

(name) \*\*\*\*\* ABNORMAL END \*\*\*\*\*

where 'name' is the name of the TL/I command being executed.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
16

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

\$GET: Input from terminal:

\$GET(buffer);


buffer: The name of a character varying string where the input from the terminal is to be placed.

RULE 1: If the data input is longer than the maximum length of the buffer, the input is truncated on the right.

Examples:

\$GET(MEMNAME);

\$GET(FI(J+K));

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page	17
	Section Name					
	TSO UTILITIES					
	Subsection Name					
Date Issued	1-20-78					
Replaces Issue	TL/I Language Reference Manual					
	Subject					
TL/I Statements						

\$LINK:      Execute a program from PL/I.

\$LINK (member,parm,clock,file,dd[,page]);

See identical parameters for \$CALL statement.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
18

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/I Statements

\$PUT, \$PUTL: Output to terminal:

```
$PUT(string);  
$PUTL(string);
```

string: An expression which must resolve to a character string. This string is output to the terminal. If \$PUTL is used, a carriage return is not issued after the string is output to the terminal; therefore, it is useful for prompting purposes.

Examples:

```
$PUT(PSCBUSER||': THE TIME IS '|| SUBSTR(TIME,1,6));
```

```
$PUT('INVALID FILE NAME;');  
$PUTL('REENTER-');
```

```
$PUTL('ENTER YOUR MOVE:');
```





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.1	Page	19
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Language Reference Manual				
Subject	TL/I Statements				

\$QUIT: Issue termination message.

\$QUIT;

No parameters.

\$QUIT will cause the following message to be displayed at the user's terminal:

(time) (name) ENDED

Where 'time' is the current clock time and 'name' is the name of the TL/1 command being executed.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
20

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

\$RCPUT: Issue RETURN CODE message.

\$RCPUT;

RULE 1: ECTRTCD must contain the return code and ECTSCMD must contain the program's or command's name.

RULE 2: TLIHIGH will be updated to contain the highestabend code or return code discovered.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
21

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject TL/I Statements

\$STAK: Add to list:


\$STAK(string);

string: An expression which must resolve to a character string. This string (which is usually a TSO command) is added to a stack of strings. If these strings are TSO commands, the stack can be thought of as an in-core CLIST.

Examples:

\$STAK('FREE FI(INPUT)');

\$STAK('ALLOC FI('|| \$DDN ||') DA('|| NAME(I,3)||') SHR');

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page	22
	Section Name	TSO UTILITIES				
	Subsection Name	TL/I Language Reference Manual				
	Subject	TL/I Statements				
Date Issued	1-20-78					
Replaces Issue						

**\$STOR:** Use the stack

**\$STOR;**

**RULE 1:** This command takes the stack built up by the use of \$STAK commands and makes it the current input/output mode. This implies:

- a) If commands like EDIT which have subcommands are issued, the subcommands will be taken from the stack.
- b) If the stack is not used up as subcommands when the TL/I command is finished, the stack then behaves exactly like a CLIST, and it is used up before going back to the input/output medium from which the TL/I command was entered (whether itself inside a CLIST, or from the terminal).

**RULE 2:** After the \$STOR command, the current stack is no longer available.

**RULE 3:** If the \$STOR command is issued more than once, the stacks given to TSO are themselves "stacked" by TSO exactly the way CLISTS within CLISTS within CLISTS are: last in, first out.

**RULE 4:** The only way the stack, when handed to TSO, is not like "EXEC"ing a CLIST is that:

- a) No substituting is allowed (no PROC statement).
- b) The rest of the commands issued in the TL/I program using the \$CMD statement are done first except as in RULE 1, a.

**Examples:**

```

$STAK('TAB ON(10, 16, 36, 41, 72)');
$STAK('TOP');
$STAK('VERIFY');
$STAK('NOTE "EDIT"');
$STAK('TENTER');
$STAK('SAVE');
$STAK('END');
$STOR;
$CMD('EDIT PROGRAM.PL1');

```



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
23

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

TL/1 provides a set of preprocessor functions available to the TL/1 user through TLICOPY. These functions are:

\$ALNUM  
\$ALPHA  
\$CAPS  
\$DDN  
\$DIGIT  
\$DSN  
\$HEX  
\$MEM  
\$TIME  
\$USTAK



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.1

Page  
24

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Replaces Issue

Subject TL/1 Functions

\$ALNUM: Only Alphanumeric

If \$ALNUM(string)=0 THEN DØ;

string: A character expression which, when resolved, will be tested for characters which are not a digit or not alphabetic. The position of the first character which is neither a digit nor alphabetic is returned. If the string is alphanumeric, it returns 0.

RULE 1: The characters @\$# are considered alphabetic.

Examples:

```
DØ WHILE ($ALNUM(NAME)>0);  
J=$ALNUM(NAME);  
$PUT('CORRECT INPUT');  
$PUTL(SUBSTR(NAME,1,J-1)); /*GOOD PART*/  
$GET(NEWPART);  
NAME=SUBSTR(NAME,1,J-1) NEWPART;  
END;
```



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
25

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$ALPHA: Only Alpha:

IF \$ALPHA(string)=0 THEN DO;

string: A character expression which, when resolved, will be tested for non-alphabetic characters. The position of the first character which is not alpha is returned. If the string is alpha, it returns a value of 0.

RULE 1: The characters @\$# are considered alphabetic.

Examples:

```
DO WHILE($ALPHA(NAME)>0|NAME="");
  $PUTL('REENTER NAME:');
  $GET(NAME);
END;
```



Standard Oil  
Company (Indiana)

Date Issued

1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
26

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

**\$CAPS:** Lower to Upper Case:

```
cstring = $CAPS(string);
```

**string:** An expression which when evaluated is a character string.

**cstring:** A string which will contain the contents of string except that all lower case alpha will be converted to upper case.

**Examples:**

```
UPPER=$CAPS(LOWER);  
$CMD('FREE FI('|| $CAPS(INPUT)||')');
```





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
27

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$DDN: Invent file name:

name=\$DDN:

name: An eight-character string of the form:

TLInnnnn

where nnnnn is a unique (for the duration of the logged on session or batch job) number.

Examples:

```
FILE=$DDN;  
$CMD('ALLOC FY(' || $DDN || ') DA(MYLIB) SHR');
```



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.1

Page  
28

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Replaces Issue

Subject TL/1 Functions

\$DIGIT: Only numeric:

IF \$DIGIT(string)=0 THEN DO;

string: A character expression which, when resolved, will be tested for non-numeric characters. The position of the first character which is not numeric is returned. If the string is numeric, it returns 0.

Examples:

DO WHILE ( \$DIGIT (TRACKS)=0);  
etc.

IF \$DIGIT (LRECL)=0 THEN  
BLOCK=LRECL\*RECNØ;



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
29

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$DSN: Return dataset name:

dataset = \$DSN(string);

string: An expression which, when resolved, is a character string of 56 characters or less. The character string should contain a dataset name of the format: dataset(member) or 'dataset(member)'.

dataset: Will contain the only dataset name that was specified in the string.

Examples:

```
LOAD=$PDSN($DSN(OBJ),'TEMPNAME');
```

```
/*CREATE FULLY QUALIFIED LIBRARY NAME IF NOT ALREADY */
```

```
IF SUBSTR($DSN(LIB),1,1) = '' THEN ;
```

```
ELSE NAME=''' || PSCBUSER || '.' || $DSN(LIB) || ''';
```



Standard Oil  
Company (Indiana)

Date issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
30

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$HEX: Display in hexadecimal:

hstring = \$HEX(number);

number: This is an integer value.

hstring: A character varying string eight characters long or less, which will contain 'number' in printable hexadecimal format with all leading zeros removed. If number =0 then hstring='0'.

Examples:

```
$PUT('THE VALUE IS' || $HEX(VAL(1)));
```

```
IF $HEX(376) = INCODE THEN etc.
```



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
31

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$MEM: Return member name

member = \$MEM(string);

string: An expression which, when resolved, is a character string of 56 characters or less. The character string should contain a dataset name of the format: dataset(member) or 'dataset(member)'.

member: Will contain the member name specified in the string or null if no member specified.

Examples:

```
OBJECT=$PDSN('MYLIB.OBJ',$MEM(SOURCE));
```

```
LIB=$PDSN($DSN(LIB),$MEM(LIB)); /* ACCOMPLISHES NOTHING */
```

```
IF $MEM(DA(1))="" THEN DO;  
  $PUTL('ENTER DA#'||1||' WITH MEMBER NAME:');  
  $GET(DA(1));  
END;
```



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
32

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$TIME: Day and Time stamp:

stamp=\$TIME;

stamp: This character string will contain a stamp similar to the one on HASP console messages. Applicable for heading up messages.

Example:

\$PUTL(\$TIME||' PLEASE ENTER CODEWORD: ');



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
33

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject TL/1 Functions

\$USTAK: Remove from stack

buffer=\$USTAK;

buffer: A character string which will be given the contents of the last entry placed on the stack. The entry placed in the buffer will no longer be on the stack.

RULE 1: If the length of buffer is less than the last entry on the stack, the contents are truncated to the right. Conversely, the string is padded with blanks to the right.

Examples:

TABLE(1)=\$USTAK;



Standard Oil  
Company (Indiana)

Date Issued 1-20-78


Replaces Issue

Manual	FSD-4	Section	97.1	Page	34
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Language Reference Manual				
Subject	On Conditions				

TL/1 initialization sets up default error recovery conditions, which can be used or overridden by the TL/1 user. The ON CONDITIONS set up includes:

ABEND  
ATTN  
RC



 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page 35
	Section Name	TSO UTILITIES			
	Subsection Name	TL/1 Language Reference Manual			
	Subject	On Conditions			
Date Issued	1-20-78				
Replaces Issue					

ABEND:      Abend condition handling.

Coding Format:

```

ØN CØNDITION(ABEND) BEGIN;

    /* ABEND HANDLING CØDE */

    END;

```

The default is:

```

ØN CØNDITION(ABEND) BEGIN;
    IF TLIBIT4='1' THEN DØ;
        SIGNAL ERROR; /* MVT BATCH */
    END;
    ELSE DØ;
        $ABPUT; /* MVT TSØ OR VS2R2 */
    END;
END;

```

RULE 1:      When ABEND is signalled, ECTRTCD contains the abend codes and ECTSCMD contains the program or command name that failed.

RULE 2:      If ECTRTCD is less than 4096, then the abend code is a user code. If ECTRTCD is greater than 4095, then ECTRTCD/4096 is the system abend code.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
36

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject On Conditions

ATTN: Attention interrupt handling.

Coding Format:

```
ON CONDITION(ATTN) BEGIN;
```

```
/* ATTENTION HANDLING CODE */
```

```
END;
```

The default is:

```
ON CONDITION (ATTN)  
$PUT('ATTENTION NOT SUPPORTED');
```

RULE 1: When the ATTN key is struck, the contents of the field TLINAME is printed out and if the user at this point enters data, the data is placed in TLIBUF. TLINAME is initially set to the name of the command and TLIBUF is initially set to blanks.

RULE 2: If, when the ATTN key is hit the user again hits ATTN, control is returned to the user and the current TL/I command is ended. If the user hits carriage return, then the TL/I command is reentered as if the ATTN key was not struck. If data is entered, it is placed in TLIFUB and the ON CONDITION is given control.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
37

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject On Conditions

RC: Return code handling:

Coding Format:

```
ON CONDITION(RC) BEGIN;
```

```
/* RETURN CODE HANDLING CODE */
```

```
END;
```

The default is:

```
ON CONDITION(RC) BEGIN;
```

```
$RCPUT;
```

```
END;
```

RULE 1: When RC is signalled, ECTRTCD contains the return code, and ECTSCMD contains the name of the program or command name that returned it.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
38

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject Include Groups

TL/1 initialization provides the TL/1 user with mappings for several control blocks/data areas which may be of use to a command processor.

These control blocks include:

TLIECT  
TLIPSCB  
TLIUPT  
TLICOPY  
TLIDATA



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
39

Section Name TSO UTILITIES

Subsection Name TL/I Language Reference Manual

Subject Include Groups

TLIECT: ENVIRONMENTAL CONTROL TABLE

%INCLUDE(TLIECT);

This control table is automatically included by TLICOPY; however, user written subroutines may need its definition. It contains:

ECTRCDF CHAR(1),	'10000000'B if abend has occurred in a program or command.
ECTRTCD BIT(24),	Contains the return code or abend codes from a program or command.
ECTPCMD CHAR(8),	Name of TL/I command.
ECTSCMD CHAR(8),	Name of last issued program or command.
ECTDDNUM BIT(24),	May be incremented by user for generating unique DDnames.
ECTUSER POINTER,	The user may use this field to point to data to be passed between programs or commands.

All fields of this table may be modified by the user.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
40

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject Include Groups

TLIPSCB: Protected Step Control Block

%INCLUDE(TLIPSCB);

This control block is automatically included by TLICOPY, however, user written subroutines may need its definition. It contains:

PSCBUSER CHAR(7), The TSO user's ID or the batch (VS2R2) Jobname.

PSCBGPNM CHAR(8), UNIT NAME for all temporary space requests.

PSCBATR 1 BIT(16), User or Job authority bits.

PSCBCPU FIXED BIN(31), How much CPU has been used.

PSCBSWP FIXED BIN(31), How much time in core.

PSCBLTIM FIXED BIN(31), How much wall time has passed.

PSCBDEST CHAR(8), Destination for SYSOUT.

PSCBRSZ FIXED BIN(31), How much core user requested.

PSCBUPXR POINTER, Address of RJE routing information.

This control block is read only.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
41

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject Include Groups


TLIUPT: User Profile Table

%INCLUDE(TLIUPT);

This control block is automatically included by TLICOPY, however, user written subroutines may need its definition. It contains:

UPTUSER CHAR(10),	The user can use this to save information between LOGON's or batch executions (VS2R2).
UPTNPRM BIT(1),	Control prompting.
UPTMJD BIT(1),	Control message headers.
UPTNCØM BIT(1),	Control inter-user communication.
UPTPAUS BIT(1),	Control CLIST pause.
UPTALD BIT(1),	Control ATTN as line delete.
UPTCDEL CHAR(1),	Character delete character.
UPTLDEL CHAR(1),	Line delete character if not ATTN.

All fields of this table may be modified by the user.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page	42
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Language Reference Manual				
	Subject	Include Groups				
Date Issued	1-20-78					
Replaces Issue						

TLICOPY: Convert PL/I to TL/I

Myprog: %INCLUDE(TLICOPY);

Myprog: The name of the main TL/I program. The very first statement of a TL/I program must be %INCLUDE(TLICOPY); and it must have a label. The label must be seven characters or less.

Example:

ALLC: %INCLUDE(TLICOPY);

Besides generating the TL/I environment, it sets up the following fields.

TLISPACE CHAR(133) INIT((133)' ');  
Which can be used as a reservoir of blanks.

TLICOMMAND CHAR(256) VARYING;  
Which contains the operands input to the program if it is a command,  
or the parm = value if it is a program.

TLIUPCMD CHAR(256) VARYING;  
TLICOMMAND with all lower case letters converted to upper case.





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.1

Page  
43

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Subject Include Groups

TLIDATA: Intracommunication Area

%INCLUDE(TLIDATA);

This area is automatically created and initialized by TLICOPY, however, user written subroutines must include it. It contains:

TLIHIGH FIXED BIN(31), The highest return code or abend code from a program or command. Is returned to the system as the TL/I programs return code.

TLICODE FIXED BIN(31), Return code from TL/I service routines (see chapter on DYNAMIC ALLOCATION).

TLIBIT1 CHAR(1), If = 'A' ATTN to be processed.

TLIBIT2 CHAR(1), If = '1' LIST(ON) was specified.

TLIBIT4 CHAR(1), If = '1' command is running in the batch (OS/MVT only).

TLIECB FIXED BIN(31), Return codes from DYNAMIC ALLOCATION.

TLICMDN CHAR(8), Name of last program or command executed.

TLIBUT CHAR(256), Data entered from ATTN.

TLINAME CHAR(8), Data typed upon ATTN.

All fields of this area may be modified by the user.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.1

Page  
44

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Language Reference Manual

Replaces Issue

Subject Sample Commands


MYTMP: /\* ALLOWS THE USER TO EXEC ALL TSO COMMANDS UNDER ANY PROCESSOR FOR WHICH  
'\$' IS A VALID SUBCOMMAND \*/

DCL MYTN BIT(1),  
STRING CHAR(256) VARYING;

ON CONDITION(ATTN) MYTN='1'B;

TLINAME=' READY'; /\* PROMPTING STRING \*/  
MYTN='C'B; /\* TLINAME OUTPUT WHEN ATTN IS HIT \*/

```
DO WHILE(STRING7='END');  
  IF MYTN THEN DO;  
    MYTN='0';  
    STRING=TLIBUF; /*TLIBUF IS INPUT WHEN ATTN */  
    TLIBUF=(256)' '; /* IS HIT */  
    END;  
  ELSE DO;  
    $PUT(TLINAME); /* PROMPT */  
    $GET(STRING); /* GET COMMAND */  
    END;  
    STRING=$CAPS(STRING);  
    IF STRING='END' then  
      $CMD(STRING); /* ISSUE COMMAND */  
    END;  
    RETURN;  
END;
```

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.1	Page 45
	Section Name	TSO UTILITIES			
	Subsection Name	TL/1 Language Reference Manual			
	Subject	Sample Commands			

Date Issued 1-20-78

Replaces Issue

/\* MY FIRST USEFUL CMOD: PROFILE \*/

COMPILE TIME MACRO PROCESSOR  
MACRO SOURCE2 LISTING

```

1  /* MY FIRST USEFUL CMOD: PROFILE */
2  PROFILE: %INCLUDE (TLICOPY);
3  DCL OUT CHAR(22) VARYING;
4  DCL IN CHAR(12) VARYING;
5  OUT='PLEASE ENTER PASSWORD';
6  $PUT(OUT);
7  $GET(IN);
8  IF IN=DATE THEN GO TO PRIME;
9  $CMD('LOGOFF');
10 RETURN;
11 PRIME;
12 $PUT('OK');
13 $CMD('FREE FI(SYSUADS)');
14 $CMD('ALLOC FI(SYSUADS) DA(''CTSO.UADS'') SHR');
15 $CMD('FREE FI(SYSPROC)');
16 $CMD('FREE FI(SYSPROC2)');
17 $CMD('ALLOC FI(SYSPROC) DA(SESSION.CLIST) SHR');
18 $CMD('ALLOC FI(SYSPROC2) DA(''CTSO.CLISTLIB'') SHR');
19 $CAT('SYSPROC','SYSPROC2');
20 $PUT(' ');
21 $PUT(' ');
22 $PUT('GOOD MORNING, MASTER');
23 $PUT(' ');
24 RETURN;
25 END;

```



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual

FSD-4

Section 97.2

Page

1

Section Name

TSO UTILITIES

Subsection Name

TL/1 Programmer's Guide

Subject

Introduction

This document, and the language it describes, is still in a state of development.

The authors will appreciate any comments or corrections. Correspondence should be sent to the address shown below.

Correspondence

Standard Oil Company (Indiana)  
Software Development Division  
200 East Randolph Drive, Mail Code 0505  
Chicago, Illinois 60601



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.2

Page  
2

Section Name TSO UTILITIES


Subsection Name TL/1 Programmer's Guide

Subject Table of Contents

Using the Language

Coding TL/1 Programs

Debugging Aids

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.2	Page	3
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Programmer's Guide				
	Subject	Using the Language				
Date Issued	1-20-78					
Replaces Issue						

A set of system %INCLUDE groups are maintained to make the standard %INCLUDE group as release independent as possible. The standard include group is called TLICOPY, and must be invoked at the front of each TL/1 program. The format of the start of a TL/1 program is shown below:

PROCEDURE-NAME: %INCLUDE (TLICOPY);

The full power of the PL/1 language is available for string manipulation, logical operations, and list processing, also the current implementation of TL/1 supports any of the I/O facilities of the PL/1 language.

Any language capable of interfacing with PL/1 can be used to write subprograms. All system supplied subprograms are stored in library U.ULMVS, which is accessible to both foreground and background users. User TL/1 programs are stored in the user's LOAD library.

NO OS SUPERVISOR SERVICES WHICH CAN CAUSE A LONG TERM WAIT should be used by any program or subprogram.


The user must refer to the dataset SYSU.UMMVS with a SYSLIB DD statement, when using the PL/1 compiler to produce TL/1 programs, and to U.ULMVS when link editing.

The language is completely described by the union of this document and the PL/1 Optimizing Compiler Language Reference Manual.

In either foreground or batch, the language may be compiled using the basic PL/1 optimizer facilities with MACRO turned on, and the compile and link steps pointing to the appropriate SYSLIB datasets as described below.

The user wishing to access session-dependent information, contained in TL/1 or TSO control blocks, is cautioned against using any DSECTS not shown in the section on include groups in the reference manual.

The functions described in this manual are currently implemented for TL/1 VERSION 3. The user must concatenate U.ULMVS with SYS1.PLIBASE at LINKEDIT time to obtain these functions. Calling sequences observe all PL/1 syntax conventions.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.2	Page	4
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Programmer's Guide				
	Subject	Coding TL/1 Programs				
Date Issued	1-20-78					
Replaces Issue						

### Coding in TL/1

Aside from special conventions required for the TL/1 interfaces and the special keywords supported by the macro preprocessor, a TL/1 program is coded in ordinary PL/1 syntax, following the conventions and restrictions in force under the PL/1 compiler in use.

### Coding Conventions

#### Margins

SORMGNS = (2,72,1) is the compiler default.  
No other margins should be specified.

#### Columns

The columnar conventions shown are optional for user TL/1 programs, but mandatory for all %INCLUDE groups, preprocessor procedures, and library service procedures.

### Margin Column Use

		Labels
A	2	
B	7	Procedures; Major Blocks; Declaratives
C	16	Minor Blocks; Statements; Minor Declaratives
D	21	Nested Statements; Statement Continuations
E	36	Comments


### Declaratives

- . Full words should be declared as fixed binary (31,0), rather than fixed (31).
- . Half words should be declared as fixed binary (15,0), rather than fixed (15).
- . Pointers should be declared at PTR; or PTR,-with NO SPACES, to facilitate changes to and from PL/S.
- . Set a field to a value rather than using "INITIAL".
- . Do not declare any "STATIC" data items.

### Coding Form

When editing a TL/1 command in the foreground, a reasonable edit tab setting would be:

```
TAB ON(7 16 21 36 65 70)
- Where the '70' is for the
  "*/" ending in comment cards.
```

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.2	Page	5
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Programmer's Guide				
	Subject	Coding TL/1 Programs				
Date Issued	1-20-78					
Replaces Issue						

### Initialization

ALL TL/1 main procedures should begin with:

- 1) A comment card with the title of the command (first card in source program).
- 2) A card of the format NAME: %INCLUDE(TLICOPY);-where NAME is the name of the TL/1 command being written.

### Naming Conventions

User written TL/1 commands must conform to the following naming conventions. Programs violating these standards may produce unpredictable results, and will be considered unsupported.

- . All names must begin with a dollar sign (\$).
- . Module/Member name must be the same as the TL/1 main procedure name. (Seven characters or less.)
- . No names may start with "SYS", "TLI", or "TLL". These prefixes are reserved for system use.

### Coding

Following the initialization portion of the program will be the executable statements comprising the command processor being written. Both command operand examination and terminal prompting are available as sources of input to the command processor. Refer to the section of the language reference manual entitled PARSING THE COMMAND BUFFER, for a description of parsing facilities available in TL/1.

The TL/1 programmer should be familiar with Type-1 TSO commands, since it is likely that some of them will be invoked through the \$CMD statement to support the command being written.

The simplest form of a TL/1 foreground command consists of operand collection (parsing and prompting) followed by a string of TSO commands issued through the \$CMD statement. The beginner may wish to convert several of his CLIST's in this fashion to gain experience in the language before attempting more complex functions.

The simplest form of TL/1 batch control program consists of parameter operand collection (parsing), with possibly Job Request card editing, followed by [, under VS2R2, dataset allocation, and] program invocations.





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual

FSD-4

Section 97.2

Page

6

Section Name

TSO UTILITIES

Subsection Name

TL/1 Programmer's Guide

Subject

Debugging Aids



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
1

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Table of Contents

Introduction

Description of TSO/TL/1 Interface

Support Routines

Introduction

Conventions for Writing Support Routines

Description of Existing Support Routines



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
2

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Introduction

This document and the language it describes, are still in a state of development. This document and IBM's PL/1 Optimizing Compiler Language Reference Manual (SC33-0009), describes the Language TL/1.

The authors will appreciate any comments or corrections. Correspondence should be sent to the address shown below.

#### Correspondence

Standard Oil Company (Indiana)  
Software Development Division  
200 East Randolph Drive, Mail Code 0505  
Chicago, Illinois 60601

#### Introduction

To use and understand the information contained in this document, familiarity with the following documents is required:

TL/1 Reference Manual	CSD0051
TL/1 Programmer's Guide	CSD0052
PL/1 Optimizer Programmer's Guide	SC33-0006
PL/1 Optimizer Language Reference	SC33-0009

This document is intended for the Systems Programmer engaged in the installation, maintenance, debugging, or enhancement of Time Sharing Language One.

The information contained herein should be used as a guide to the actual code. This document is intended to function as a guide to the source code, not as a complete, independent, logic description.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
3

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject System Interfaces

The basic interface between TL/1 commands and the system is the command processor \$, which receives control when a TL/1 command is invoked. \$ follows the normal OS/VS conventions for command processors. It searches for the appropriate TL/1 module (as specified by 'NAME' in command) and loads it. After building and/or copying control blocks, \$ passes control to the TL/1 module via ATTACH.

TL/I command processor modules establish the environment necessary to process commands and interact with TSO by executing routine TLINIT, which is invoked immediately upon entry to a TL/I main procedure. The call to this routine and other routines and data structures which define control blocks and parse the command input are included in a TL/1 program by the PL/1 preprocessor statement:

```
% INCLUDE (TLICOPY)
```

Support routines called by TL/I command processors may be written in BAL, but must follow TL/I conventions for register usage. See section 'Support Routines.'



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

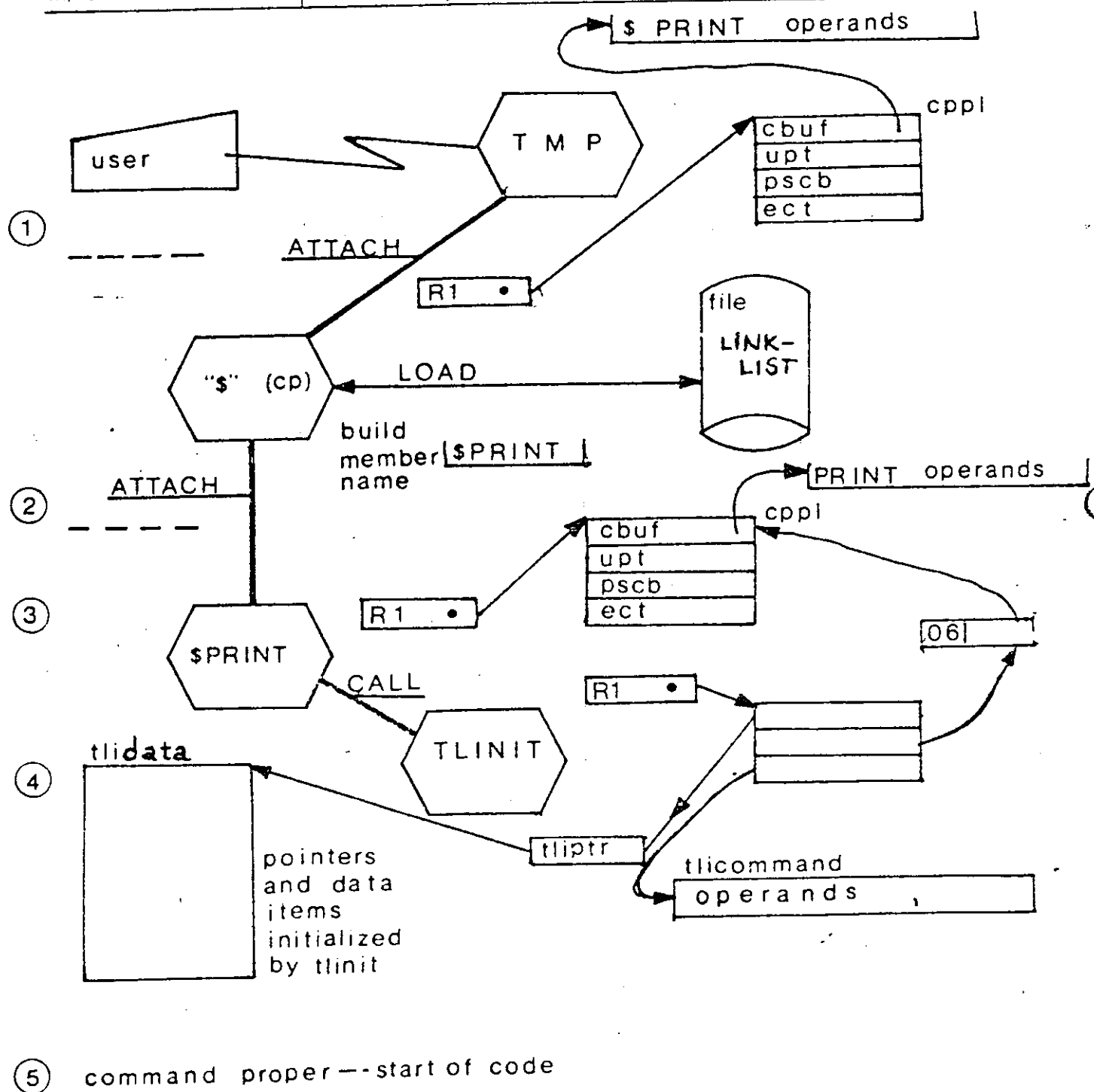
Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject System Interfaces

Section 97.3

Page  
4



TL / 1 INVOCATION



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.3

Page  
5

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Replaces Issue

Subject System Interfaces

Note to diagram 'TL/1 INVOCATION'

1. Command processor '\$' is attached by the TMP upon receipt of command from TSO user.
2. The appropriate TL/1 command processor is attached by '\$' (in this example, \$PRINT).
3. Upon entry to \$PRINT, routine TLINIT is invoked; this routine initializes the TL/1 environment. (Note-the call to TLINIT is part of the TL/1 initialization processing performed by code resulting from % INCLUDE (TLICOPY).)
4. Upon completion of TLINIT, pointers and data items in the area TLIDATA are initialized and available to the TL/1 module.
5. Other routines included in member TLICOPY set up 'ON CONDITIONS' for default error recovery and define commonly used control blocks. Code for TL/I functions is included as preprocessor blocks; invocations of these functions by the TL/1 module are thus replaced by the preprocessor with the code they generate, expanding in the TL/1 module as an assembly language macro would expand into assembly language source. These functions include routines to parse the command buffer.

### Parsing

A unique syntax is supported by TL/1, allowing the user to enter parameters as keyword or positional. To support this facility, the PL/1 macro facility is used to allow definition of the operands, the order of appearance in a positional string, an abbreviation for keyword entry, and characteristics of the operand.

The in-line parse code generated by the macros, interfaces with the TLIARRY routine which performs scanning of values within lists.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
6

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Support Routines

## Introduction

The support routines used by TL/1 are the foundation of the language, and provide much of the "power" available to the TL/1 programmer. The macro language and the support routines are interrelated and interdependent. Wherever possible, conventions specified in the PL/1 Programmer's Guide have been adhered to, so as to make the TL/1 language as release-independent (and compiler-independent) as possible.

## Writing An Assembler Support Routine

Assembler support routines must be written using the TL1START and TL1STOP macros, and must observe the conventions described here. Before attempting to write an assembler support routine, the source for several existing support modules should be examined to provide background knowledge about conventions and techniques.

### Support Routine

#### Register Usage

R0	Parameter Register
R1	Parameter List Register
R2	Work Register
R3	Work Register
R4	Work Register
R5	Work Register
R6	Work Register
R7	Work Register
R8	Work Register
R9	Reserved
R10	Pointer to TLIWORK
R11	Base Register
R12	PL/1 TCA
R13	PL/1 DSA
R14	Return Register
R15	Entry Point/Return Code Register

## Writing A PL/1 Support Routine

Support routines written on PL/1 must conform to the normal PL/1 conventions for non-main called procedures. Any support routine, using other TL/1 support functions, should be passed 'TL1PTR' as its first input argument, and a %INCLUDE for "TL1DATA" should appear after the procedure statement of the routine being written.

Where possible, support routines returning a single argument should be written and declared as functions.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.3

Page  
7

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Replaces Issue

Subject Description of Existing Routines

The routines listed below are the support routines used by the TL/1 language macros to perform their functions. An "external" routine is one which interfaces with the Operating System, or performs a function with a scope outside the TL/1 program. An "internal" routine provides a function or service which affects only action/data internal to the TL/1 program.

Routine	TL/1 Macro	ASM	PL/1	Internal	External
TLIARRY	\$PARSE		X	X	
TLIATTN	-NONE-	X			X
TLICALL	\$CALL	X			X
TLICAPS	\$CAPS		X	X	
TLICAT	\$CAT	X			X
TLICOPY	-NONE-	X			X
TLICP	\$CMD	X			X
TLIDEL	\$DELETE	X			X
TLIDSN	\$DSN		X	X	
TLIDSX	-NONE-	X			X
TLIGET	\$GET	X			X
TLIHEX	\$HEX		X	X	
TLILINK	-NONE-	X			X
TLILNK	\$LINK	X			X
TLILOAD	\$LOAD	X			X
TLIMEM	\$MEM		X	X	
TLINIT	-NONE-	X			X
TLIPDQ	-NONE-	X			X
TLIPDSN	\$PDSN		X	X	
TLIPUT	\$PUT,\$PUTL	X			X
TLISTAI	-NONE-	X		X	
TLISTAK	\$STAK	X		X	
TLISTOR	\$STOR	X			X
TLITIME	\$TIME	X		X	
TLIUSTAK	\$USTAK	X		X	
TLIXIT	-NONE-	X			X





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.3	Page	8
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Routine: TLIARRY

Macro: \$PARSE

Language: PL/1

Function:

The TLIARRY routine supports the TL/1 parse facility. It performs the actual parsing of all list operands.

External References:

Input command strings, control tables, and destination variables in the main TL/1 program are accessed by TLIARRY.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
9

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLIATTN

Macro: NONE

Language: ASSEMBLER

Function:

The TLIATTN routine accepts the attention interrupts from TSO and causes the raising of the "ATTN" On-Condition within the TL/1 main program.

External References:

This routine uses the PL/1 "SIGNAL CONDITION" routine to cause the ATTN condition to be raised.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.3	Page	10
Section Name		TSO UTILITIES			
Subsection Name		TL/1 Program Logic and Data Structures			
Subject		Description of Existing Routines			

Routine: TLICALL

Macro: \$CALL

Language: ASSEMBLER

Function:

The TLICALL routine invokes a program in a manner similar to "EXEC PGM=" in the batch. Parameter string, OS/Utility DDNAME Lists, and time limitation (322 clock) are supported.

External References:

The user's TQE.

The linklist.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
11

Section Name TSO UTILITIES

Subsection Name: TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLICAPS

Macro: \$CAPS

Language: ASSEMBLER

Function:

The TLICAPS routine translates all lower case to upper case. Only EBCDIC is supported.

External References:

Only operands passed by the caller.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.3

Page  
12

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Replaces Issue

Subject Description of Existing Routines

Routine: TLICAT

Macro: SCAT

Language: ASSEMBLER

Compatible with MVS: NO

Function:

The TLICAT routine calls the DAIR routine to concatenate up to fifteen datasets (by filename/ddname). The DAIR control blocks are received and freed by this routine each time it is invoked.

External References:

IKJDAIR and DAIR Control Blocks.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
13

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLICOPY

Macro: NONE

Language: ASSEMBLER

Function:

TLICOPY is the TL/1 Copy/Reformat Utility.

External References:



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
14

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLICP

Macro: \$CMD

Language: ASSEMBLER

Function:

The TLICP interface provides the facility to attach a TSO command processor in a manner analogous to the TMP.

External References:

CPPL, PSCB, ECT, UPT: command/operand string passed by caller.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.3	Page	15
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Routine: TLIDEL

Macro: \$DELETE

Language: ASSEMBLER

Function:

TLIDEL is the TL/1 Module Delete Interface.

External References:





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
16

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLIDSN

Macro: \$DSN


Language: PL/1

Function:

The TLIDSN routine extracts the dataset name portion of a PDS-member  
dsname.

External References:

The input string passed by the caller.

 Standard Oil Company (Indiana)	Manual	FSD-4	Section	97.3	Page	17
	Section Name	TSO UTILITIES				
	Subsection Name	TL/1 Program Logic and Data Structures				
	Subject	Description of Existing Routines				

Date Issued 1-20-78

Replaces Issue

Routine: TLIDSX

Macro: NONE

Language: ASSEMBLER

Function:

This module (used by \$PMAP and \$COPY) reads the Format 1 DSCB for a requested dataset, formats requested data, and passes this formatted data and the DSCB back to the caller.

External References:

DSCB1



Standard Oil  
Company (Indiana)

Date Issued

1-20-78

Replaces Issue

Manual

FSD-4

Section

97.3

Page  
18

Section Name

TSO UTILITIES

Subsection Name: TL/I Program Logic and Data Structures

Subject

Description of Existing Routines

Routine: TLIENQ#

Macro: NONE

Language: ASSEMBLER

Function:

TLIENQ# is called by \$WHO to retrieve and format enqueue information for a specified dataset. TLIENQ# obtains major and associated minor QCB's and constructs a table with information from the QCB's and related QEL's and CESD's. Routine TLIENQ2# is then called to do I/O.

External References:

QCB - Queue Control Block

QEL - Queue Element

CSCB - Command Schedule Control Block



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.3

Page  
19

Date Issued 1-20-78

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Replaces Issue

Subject Description of Existing Routines

Routine: TLIENQ2#

Macro: NONE

Language: ASSEMBLER

Function:

This routine formats enqueue information retrieved by TLIENQ#, does a TPUT, and frees all work areas obtained by TLIENQ#.

External References:



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.3	Page	20
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Routine: TLIGET

Macro: \$GET

Language: ASSEMBLER

Function:

The TLIGET routine retrieves a line on input from the terminal using  
-TGET-.

External References:



Standard Oil  
Company (Indiana)

Manual	FSD-4	Section	97.3	Page	21
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Date Issued 1-20-78

Replaces Issue

Routine: TL1HEX

Macro: \$HEX

Language: PL/1

Function:

The TL1HEX routine converts a full-word binary field to eight printable hexadecimal digits, suppressing leading blanks.

External References:

The input field passed by the caller.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.3

Page  
22

Date Issued 1-20-78

Section Name TSO UTILITIES

Replaces Issue

Subsection Name TL/I Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLIKEYS

Macro: NONE

Language: PL/I

Function:

TLIKEYS parses keyword definitions specified by coding \$KEY and \$KEYDEF functions in a TL/I program. TLIKEYS also sets up bit matrices which will describe the presence or absence of a particular \$KEY keyword in a given command.

External References:



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 98.3

Page  
23

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLILINK

Macro: -NONE-

Language: ASSEMBLER

Function:

TLILINK does a BALR 14,15 (assuming LOAD and registers already set).

External References:





Standard Oil  
Company (Indiana)

Manual	FSD-4	Section	97.3	Page	24
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Date Issued 1-20-78

Replaces Issue

Routine: TLILNK

Macro: \$LINK

Language: ASSEMBLER

Function:

TLILNK will issue a LINK for the programs specified in the \$LINK macro, and in effect, cause execution of the specified program from PL/I.

External References:



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
25

Section Name TSO UTILITIES

Subsection Name TL/I Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLILOAD

Macro: \$LOAD

Language: ASSEMBLER

Function:

TLILOAD is the TL/I Module Load Interface.

External References:



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.3	Page	26
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Routine: TLIMEM

Macro: \$MEM

Language: PL/1

Function:

The TLIMEM routine extracts the PDS member name from a PDS-format DSNAME, returning the member name as a Char varying string.

External References:

The input string passed by the caller.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
27

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLINIT

Macro: NONE

Language: ASSEMBLER

Function:

The TLINIT routine initialized the TL/1 portion of the PL/1 environment, including setting up of the TLIATTN routine to handle attention interrupts, initialing pointer variables, and moving the command operand string into the operand buffer in the TL/1 main program.

External References:



Standard Oil  
Company (Indiana)

Manual	FSD-4	Section	97.3	Page	28
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Date issued 1-20-78

Replaces Issue

Routine: TLIPDQ

Macro: NONE

Language: ASSEMBLER

Function:

TLIPDQ is called by \$PDS to access and manipulate the directory and selected members of a specified PDS.

External References:

The following control blocks are read by TLIPDQ:

DCB  
DEB  
DSCB  
UCB



Standard Oil  
Company (Indiana)

Manual	FSD-4	Section	97.3	Page	29
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Date Issued 1-20-78

Replaces Issue

Routine: TLIPDSN

Macro: \$PDSN

Language: PL/1

#### Function:

The TLIPDSN routine builds a PDS-member format DSNAME from a dsname and a member name. Both quoted and TSO format dsnames are handled.

#### External References:

The input strings passed by the caller.



Standard Oil  
Company (Indiana)

Date Issued

1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
30

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLIPUT

Macro: \$PUT,\$PUTL

Language: ASSEMBLER

Function:

The TLIPUT routine displays a line at the terminal using -PUTLINE-.  
An alternate entry to suppress cursor movement/carrier return is also  
supported.

External References:



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
31

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLISTAI

Macro: NONE

Language: ASSEMBLER

Compatible with MVS: NO

Function:

The TLISTAI routine provides STAI/ESTAI support for TL/1 commands. A user 322 abend is issued for a-detach-abend code. All other abends cause the retry facility to be invoked or bypassed, on the basis of code settings.

External References:

The STAE/STAI work area.





Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual	FSD-4	Section	97.3	Page	32
Section Name	TSO UTILITIES				
Subsection Name	TL/1 Program Logic and Data Structures				
Subject	Description of Existing Routines				

Routine: TLISTAK

Macro: \$STAK

Language: ASSEMBLER

Compatible with MVS: NO

Function:

The TLISTAK routine causes a string to be placed on the internal TL/1 stack.

External References:

None.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
33

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLISTOR

Macro: \$STOR

Language: ASSEMBLER

Compatible with MVS: NO

Function:

The TLISTOR routine causes the current contents of the TL/1 internal stack to be placed in the TSO Command Stack.

External References:

IKJSTCK and STACK.  
Control Blocks.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
34

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLITIME

Macro: \$TIME

Language: ASSEMBLER

Function:

The TLITIME routine returns a HASP-format time stamp.

External References:

The TIME SVC.



Standard Oil  
Company (Indiana)

Manual FSD-4

Section 97.3

Page  
35

Date Issued 1-20-78

Replaces Issue

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLIUSTAK

Macro: \$USTAK

Language: ASSEMBLER

Compatible with MVS: NO

Function:

The TLIUSTAK routine causes the last (most recent) entry in the TL/1 internal stack to be removed from the stack, and its contents returned to the caller.

External References:

None.



Standard Oil  
Company (Indiana)

Date Issued 1-20-78

Replaces Issue

Manual FSD-4

Section 97.3

Page  
36

Section Name TSO UTILITIES

Subsection Name TL/1 Program Logic and Data Structures

Subject Description of Existing Routines

Routine: TLIXIT

Macro: NONE

Language: ASSEMBLER

Function:

TLIXIT performs standard QSAM DCB exit processing.

External References:

JFCB  
DSCB