Logical Console Operator

Version 3.3.00

NETEC International, Inc.
P O Box 180549
Dallas, Texas 75218-0549
Telephone - (214) 343-9744
Telefax - (214) 343-9009
Email: support@netec2000.com

This page intentionally blank

TABLE OF CONTENTS

Introduction

LOCO, the Logical Console Operator, reduces and simplifies people's interactions with MVS, JES, NCCF, VTAM, VSAM, CICS, TSO, IMS, security packages, batch and on-line applications and hardware /software errors. When LOCO detects a system activity (triggering event) that you consider important, it precisely follows your instructions. LOCO's accuracy and reliability ultimately benefit your information center's end users. Improved on-line response, increased batch throughput, and 24 X 7 uptime are all realistic goals.

LOCO supports your operations staff with job scheduling and automatic response to WTO(R)s. It supports your network controllers by monitoring and restarting VTAM lines and devices and by automating NCCF commands and responses. It supports development staffs by translating cryptic error messages and by routing explanatory messages to TSO users. It supports technical support teams by restarting failed software components like CICS or IMS regions. LOCO routes concise messages to specific consoles and TSOids with responsibility lists and suggested corrective actions.

LOCO's role in the data center is often determined by the operations staff, the system programming staff and the network control group. Together they determine which system and application activities will be monitored and controlled by LOCO. These automatic activities include: (1) console actions triggered by WTOs, (2) console actions triggered by WTORs, (3) TIME initiated actions and (4) actions triggered by the execution of additional job steps which post messages directly to LOCO. In the course of managing resources, LOCO will submit job streams, perform complex sequences of actions specified in its control tables, or interface to other NETEC International products such as the CICS Application File Control facility and the DBControl Batch Interface program.

If LOCO is started early in your IPL sequence, it will oversee all of your procedures, including JES initialization. Conversely, LOCO will gracefully shut down NCCF and JES for scheduled IPLs or table maintenance with one command. LOCO never sleeps, never forgets, and never comes in late. It always remembers what to do for each shift no matter what day it is.

Because LOCO dramatically reduces a data center's operating complexity, less skilled people can adequately manage day-to-day operations. Since LOCO centralizes and qualifies system activity, a small staff can manage a large center, even in the shadow of growing workloads. LOCO frees your operations and technical staff members from unnecessary and tedious console monitoring so they can concentrate on more important activities.

The principle features of LOCO include:

- A syslog analysis program to suggest possible candidates for automatic replies or actions.
- A simple "fill in the blanks" language to specify triggering events and actions.
- Automatic replies to WTORs, WTOs, and WTPs.
- Automatic actions at specific times of the day.
- Wildcard characters and message text parsing.
- Multiple actions with conditional waits either for time periods or the occurrence of other events.
- Easy to use commands to control LOCO's operation.
- Open architecture with User Exits to implement complex decisions and actions.
- Job stream submission.
- Automatic or operator controlled load of new LOCO control tables.
- Automatic WTO processing with rerouting, message suppression, non-deletion, and message expansion.
- User defined operator commands.
- Extremely low overhead, no I/O to access the control tables, minimal CSA usage, optimized instruction paths for hits and misses.
- Variable symbols to insert jobname, tsoid, WTOR reply numbers, time and other system generated constants in the ACTION commands.
- A simulation mode is available which causes LOCO actions to be written to a data set rather than issued as console commands.
- Seamless interface to other NETEC International products such as CICS Application File Control (CAFC) and the DBControl batch interface.
- Secure TSO interface that provides:
  - Online table preparation (edit/browse/reload)
  - Current LOCO status display
  - Control of LOCO internal trace functions
  - Request LOCO submit batch jobs from the job submission library.
  - Manipulate the LOCO work queues. Individual queue entries may be held, released or deleted. Command queue entries may also be executed.

- Provide control of LOCO internal resources.

Operational Overview

LOCO is driven by tables of triggering events and action statements which reside in a PDS. The activity statements are coded in a source language similar in syntax to IDCAMS. The source statements are compiled into object tables. The table(s) are loaded into LOCO's private address space and become totally resident. There are nine major statements:

- DEFINE to specify global information of interest to LOCO.
- STARTUP to specify actions to occur as LOCO begins execution.
- WTO to specify actions to occur whenever a certain WTO is issued.
- EVENT to specify actions to occur whenever a certain LOCO defined event occurs.
- WTOR to specify replies to respond to certain WTORs.
- @ to specify actions to occur at a certain time of day.
- COMMAND to specify actions to occur whenever a certain console command is keyed.
- CAFCCMD to define CAFC commands that can be used as ACTION targets of other LOCO commands.
- DBCTCMD to define DBControl commands that can be used as ACTION targets of other LOCO commands.

The general setup of tables for use by LOCO can be accomplished either through batch on TSO/ISPF.

The general setup procedure for setup of LOCO through batch has 3 steps.

1. Code a LOCO control table following the statement syntax rules.
2. Run the LOCO table builder program (LOCOBLDR) to produce an object LOCO control table.
3. Execute LOCO using the control table. You may have many different LOCO control tables, perhaps for different work shifts or different days of the week. You may manually or automatically change the control table(s) currently in use at any time.

The general setup procedure for setup of LOCO through TSO has 3 steps.

1. Code a LOCO control table following the statement syntax rules through the TSO/ISPF LOCO table maintenance screens.
2. The LOCOBLDR process will be accomplished as part of the interactive table development process. This will produce an object LOCO control table.
3. Execute LOCO using the control table or ask LOCO to load the new table through the TSO/ISPF interface. You may have many different LOCO control tables, perhaps for different work shifts or different days of the week. You may manually or automatically change the control table(s) currently in use at any time.

The following is a sample LOCO control table. The table would be compiled by LOCOBLDR to check the syntax and to produce an optimized object table suitable for processing by LOCOMAIN.

```
DEFINE WILDCARD(*) PREFIX(@)
STARTUP ACTION (S TSO)
WTO (IEE362A) ACTION (S SMFDUMP) SLEEP (:30:00)
WTOR (IEF238D) REPLY (CANCEL)
@(08:15:00) ACTION (@>LOGRECA) WAITIME(:02:00)
            ACTION (@>LOGRECB)
COMMAND (@D***) ACTION (D U,,,***,1)
```

- The DEFINE card tells LOCO that an '*' is the wildcard character to be used for this table. The '@' is the prefix for LOCO control commands and for site developed custom console commands.
- The STARTUP card tells LOCO to issue the MVS command 'S TSO' only when LOCO is first initialized.
- The WTO card instructs LOCO to:
  1. issue 'S SMFDUMP' upon the occurrence of a WTO with text of 'IEE362A', and
  2. put this particular table entry to SLEEP for thirty minutes (i.e., ignore subsequent IEE362A messages for the next thirty minutes).
- The WTOR card instructs LOCO to interrogate each WTOR passing through the MVS system for the leading characters 'IEF238D'. Whenever these characters are found, LOCO will issue the MVS command 'REPLY xx,CANCEL' where xx is the appropriate reply number.
- The @ card instructs LOCO to:
  1. wait until the next clock time of 08:15 and issue the MVS command 'S LOGRECA', and
  2. then wait for two minutes before issuing the MVS command 'S LOGRECB'.

- The COMMAND card instructs LOCO to watch for a command keyed on the MVS console beginning with '@D' followed by any three characters. Whenever the command is keyed, LOCO will issue the MVS command 'D U,,,***,1' with the *** replaced by the corresponding characters in the keyed command (e. g., @D285 would produce D U,,,285,1).

Processing Narrative

LOCO is initialized at IPL time by the standard MVS subsystem mechanism controlled by the IEFSSN member of PARMLIB. The program module LOCOINIT is invoked for each LOCO subsystem defined in IEFSSN. The amount of CSA used by LOCO is determined via the user start up parameters entered via the PARM field on the LOCOMAIN execution statement. There is a minimum amount of both below the line and above the line CSA necessary for LOCO to execute. The LOCO system maintains only key control information in CSA, items such as its active/ de-active indicators, counters, queue headers and subsystem exit logic are held in this storage. The location of WTO blocks that are waiting to be processed is determined by the DS=xxx parameter found on the LOCOMAIN execute PARM. If DS=YES is specified, the WTO blocks are stored in a data space created by LOCOMAIN at startup time and no CSA is utilized. If DS=NO is specified the WTO blocks are stored in below the line CSA. In both cases the maximum number of WTOs to be held for processing is specified in the LOCO table via the QLIMIT parameter on the DEFINE command. If the WTO blocks are housed in CSA, LOCOMAIN will adjust the QLIMIT value downward if the possible usage of CSA by WTO blocks exceeds 50 percent of the available CSA at LOCO table load time. The static amount of CSA used by LOCO 3.3 is less than 4K bytes below the line and approximately 12K bytes above the line. The variable amount (only used when DS=NO is specified) is 598 (the size of one WTO block) times the current number of WTOs waiting for processing by LOCOMAIN: The max variable amount of CSA would be 1024 * 548 = 612352 bytes if the QLIMIT is equal to 1024.

LOCO activities are controlled by a table of statements specifying certain actions to be executed upon the occurrence of specific WTO(R) text, WTP text, pseudo WTO text, console command text or time of day. The actions consist of MVS console commands or LOCO built-in commands. The execution of the actions may be constrained by filters such as date, time and jobname. The action text may be modified in real time by certain variable symbol values and by direct variable substitution from the triggering WTO(R) text.

The source LOCO table statements are compiled by program module LOCOBLDR into object form usable by LOCOMAIN. LOCOBLDR performs checking for statement syntax, combinatorial correctness and completeness of operands, and structure and value ranges for operand values. The statements and operands are organized and ordered for structured processing by LOCOMAIN. Instruction path lengths for "MISSES" and "HITS" are minimized even in LOCO's no I/O environment. All statement and operand keywords are translated into tokens so that no interpretive processing is required in real time.

The heart of LOCO processing is program module LOCOMAIN. It may be invoked as a started task or as a batch job. Initially LOCOMAIN will load the specified LOCO control table into its own address space. No further I/O is required to access and process the LOCO table statements. Each statement in the control table is placed on a wait queue depending on the statement type. WTO and WTOR statements are placed on a TEXT queue waiting for a matching text to be passed from an LOCO exit, LOCO's pseudo WTO interface(LOCOINTF).
User defined and built-in LOCO command statements are placed in a COMMAND queue waiting area in LOCOMAIN while waiting for matching commands. Time initiated statements are placed on a TIME queue in ascending time of day sequence waiting for the proper time of day to occur. Startup statements are placed on a unique STARTUP queue waiting for LOCOMAIN to complete initialization. Event statements are placed on an EVENT queue waiting for notification from event processors. After all the table statements have been placed on their respective queues, a stimer is issued for the top statement on the TIME queue, and the STARTUP queue is processed.

Each STARTUP statement is removed from the queue and the filter operands are validated. The ACTIONs are executed in order with MVS commands being processed by SVC 34. Built-in LOCO commands are processed internally. If a WAITEXT operand is encountered, the table statement is placed on the TEXT queue waiting for a matching WTO. If a WAITIME operand is encountered, the table statement is placed on the TIME queue waiting for the time to expire. When all the ACTIONs of the startup statement have been processed the statement is discarded.

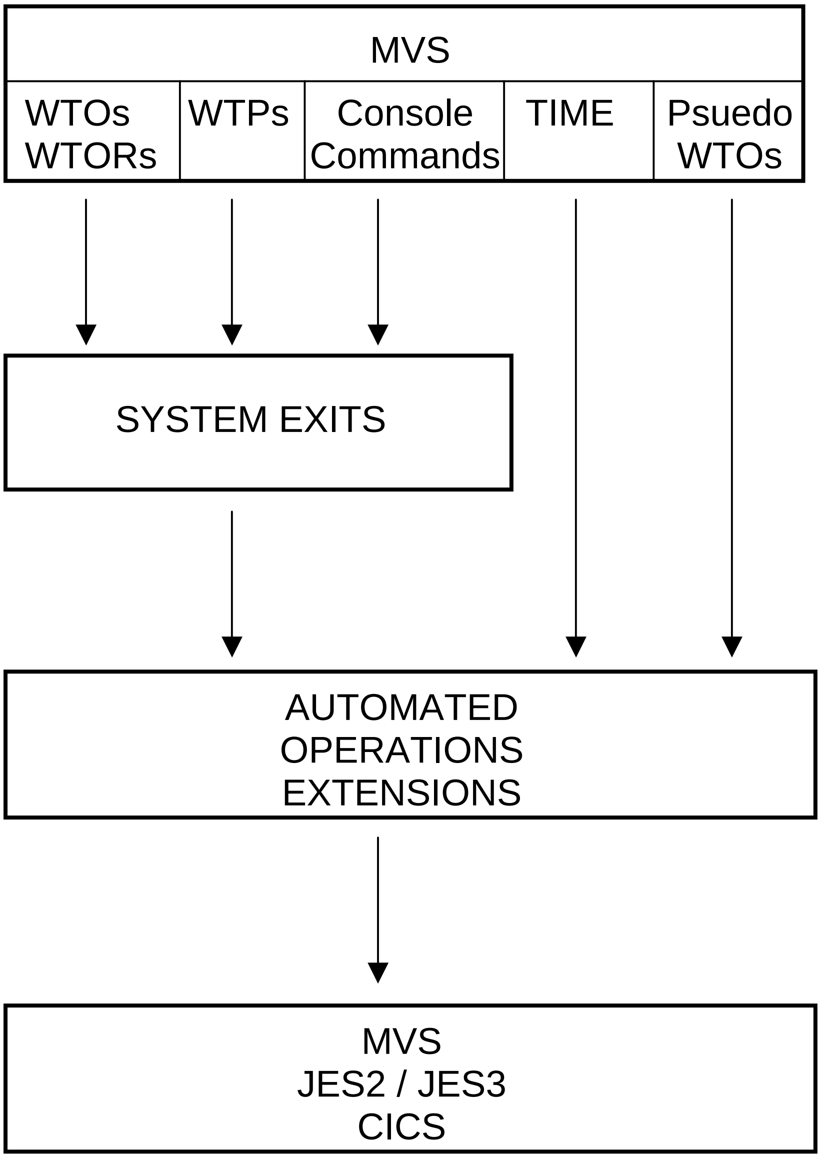
The LOCO exits and interfaces all pass text to LOCOMAIN, whether the text originates from a WTO(R), a console command or from input generated by the LOCO pseudo WTO interface, LOCOINTF or from transient data processor LOCOTDEX.

When LOCOMAIN receives TEXT from its exits or interfaces, it matches the text to the LOCO table statements on the appropriate queue. e.g. TEXT queue for a WTO(R) text or COMMAND queue for a console command. If a match is found the filter operands are validated. If the filter fails the state- ment is placed back on its original queue. If the filter passes, the ACTIONs are executed in order with MVS commands being processed by SVC 34, and built-in LOCO commands being processed internally. If a WAITEXT operand is encountered, the table statement is placed on the TEXT queue waiting for a matching WTO. If a WAITIME operand is encountered, the table statement is placed on the TIME queue waiting for the time to expire. When all the ACTIONs of the statement have been processed the statement is placed back on its original queue.

When LOCOMAIN is posted by the expiration of the stimer, each LOCO table statement on the TIME queue is processed until one is reached which is beyond current time of day. When all the ACTIONs of the time statement have been processed the statement is placed back on TIME queue at a time of day determined by the REPEAT operand.

When LOCOMAIN is requested to shutdown through a LOCO @STOP command or a MVS CANCEL or STOP command, all CSA area is freemained. LOCOMAIN then gracefully terminates.

Functional Diagram



Operating System handles messages, commands, requests, timer services, and LCON’s jobstep direct post pseudo WTOs

Standard systems and sub-systems exits interrogate message content

LCON matches messages and events to user defined tables and issues commands

Operating system processes user commands

Installation Guide

Distribution Tape

The LOCO distribution tape contains three standard labeled files:

FILE	DATA SET NAME	DESCRIPTION
1	LOCO.LOAD.LIBRARY	LOCO program load modules
2	LOCO.SOURCE.LIBRARY	LOCO installation JCL, run JCL and sample tables.
3	LOCO.ISPF.ISPPLIB	LOCO ISPF Panel library.
4	LOCO.ISPF.ISPMLIB	LOCO ISPF Message library.

Installation Overview

Step 1. Load the four distributed PDS files to your disk.

Step 2. Considerations for the upgrade installation of LOCO.

Step 3. Determine the authorization technique to be used by LOCO.

Step 4. Make the LOCOINIT program available at IPL.

Step 5. Define LOCO as a subsystem to MVS.

Step 6. Install the LOCO TSO interface.

Step 7. Define security for the LOCO TSO Interface.

Step 8. Print and review the sample LOCO control tables.

Step 9. Define LOCO control table.

Step 10. Optionally add LOCO to the installation’s MVS Automatic Restart Manager (ARM) policy.

Installation Detail

Step 1. Load the four distributed PDS files to your disk using IEBCOPY.

```
//LOCOCOPY   JOB   .................
//FILE1      EXEC  PGM=IEBCOPY
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    DSN=LOCO.LOAD.LIBRARY,
//                 DISP=OLD,
//                 UNIT=TAPE,
//                 VOL=SER=xxxxxx,    <= insert LOCO tape volser
//                 LABEL=(1,EXPDT=98000)
//SYSUT2     DD    DSN=your.load.pds,
//                 DISP=(NEW,CATLG,DELETE),
//                 UNIT=SYSDA,
//                 VOL=SER=vvvvvv,
//                 SPACE=(CYL,(2,1,10),RLSE),
//                 DCB=(RECFM=U,BLKSIZE=6144)
//SYSUT3     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN      DD    *
 COPY I=SYSUT1,O=SYSUT2
//*-----------------------------------------
//FILE2      EXEC  PGM=IEBCOPY
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    DSN=LOCO.SOURCE.LIBRARY,
//                 DISP=OLD,
//                 UNIT=TAPE,
//                 VOL=SER=xxxxxx,    <= insert LOCO tape volser
//                 LABEL=(2,EXPDT=98000)
//SYSUT2     DD    DSN=your.source.pds,
//                 DISP=(NEW,CATLG,DELETE),
//                 UNIT=SYSDA,
//                 VOL=SER=vvvvvv,
//                 SPACE=(CYL,(5,1,15),RLSE),
//                 DCB=(RECFM=FB,LRECL=80,BLKSIZE=23440)
//SYSUT3     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN      DD    *
 COPY I=SYSUT1,O=SYSUT2
//FILE3      EXEC  PGM=IEBCOPY
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    DSN=LOCO.ISPF.ISPPLIB,
//                 DISP=OLD,
//                 UNIT=TAPE,
//                 VOL=SER=xxxxxx,    <= insert LOCO tape volser
//                 LABEL=(3,EXPDT=98000)
//SYSUT2     DD    DSN=your.ispf.ispplib,
//                 DISP=(NEW,CATLG,DELETE),
//                 UNIT=SYSDA,
//                 VOL=SER=vvvvvv,
//                 SPACE=(TRK,(5,1,15),RLSE),
//                 DCB=(RECFM=FB,LRECL=80,BLKSIZE=23440)
//SYSUT3     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN      DD    *
 COPY I=SYSUT1,O=SYSUT2
//FILE4      EXEC  PGM=IEBCOPY
//SYSPRINT   DD    SYSOUT=*
//SYSUT1     DD    DSN=LOCO.ISPF.ISPMLIB,
//                 DISP=OLD,
//                 UNIT=TAPE,
//                 VOL=SER=xxxxxx,    <= insert LOCO tape volser
//                 LABEL=(4,EXPDT=98000)
//SYSUT2     DD    DSN=your.ispf.ispmlib,
//                 DISP=(NEW,CATLG,DELETE),
//                 UNIT=SYSDA,
//                 VOL=SER=vvvvvv,
//                 SPACE=(TRK,(5,1,15),RLSE),
//                 DCB=(RECFM=FB,LRECL=80,BLKSIZE=8800)
//SYSUT3     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4     DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN      DD    *
 COPY I=SYSUT1,O=SYSUT2
//*-----------------------------------------
```

Step 2. Considerations for the upgrade installation of LOCO.

If this is a new installation of LOCO this step should be bypassed. If you are upgrading from a previous version of LOCO there are several items that should be considered during the installation process. The following is a brief list of these items.

- ***All LOCO tables must be rebuilt using the LOCOBLDR utility because the compile LOCO table format has changed in this release of LOCO.***
- The LOCO SVC does not have to be refreshed if it is already install from a previous LOCO installation.
- Care should be taken when placing the LOCOINIT program into the system link list. The versions of LOCOINIT are not backward compatible. Once the copy of LOCOINIT has been placed in the system link list and either an IPL or link list refresh has been accomplished, older versions of LOCO will not initialize. The following technique is recommended for testing new releases of LOCO when previous versions are active in the system.

  - Execute LOCOINIT as a batch job to initialize one of the LOCO subsystem names to be used by this release of LOCO.
  - Execute LOCOMAIN as a batch job to test it’s functionality.
  - Once the new version of LOCO has been tested, make the appropriate changes to the system link list to activate the use of the new version.

Step 3. Determine the authorization technique to be used by LOCO.

Many of the LOCO programs require APF authorization to function correctly. There are two methods to achieve the required authorization. LOCO can be executed from an MVS authorized library or a LOCO supplied authorization SVC can be installed at your installation. The following contain information about each of these methods.

Method 1 Authorization of the LOCO program library.

Either adding the LOCO library to the MVS link list or placing the LOCO library in the authorized library list in SYS1.PARMLIB can authorize the LOCO program library.

To add the LOCO libraries to the MVS link list, the current link list definition, usually found in the PROGxx member of SYS1.PARMLIB, must be modified. The following example shows a partial listing of the link list portion of a PROGxx member before the LOCO library has been added. The exact format of the LNKLST card should be verified in the Initialization and Tuning Guide for the MVS version that is executing in your installation.

```
LNKLST DEFINE NAME(LNKLST00)
LNKLST ADD NAME(LNKLST00) DSN(SYS1.SERBLINK)
LNKLST ADD NAME(LNKLST00) DSN(NETV130.SCNMLNK1)  VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(NETV130.SEKGMOD1)  VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(NETV130.SEKGMOD2)  VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(IGY210.SIGYCOMP)   VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(FAN130.SEAGALT)    VOLUME(Z1RES2)
    ..
    ..
LNKLST ACTIVATE NAME(LNKLST00)
```

The following example is a partial listing of the same PROGxx member with the LOCO library added.

```
LNKLST DEFINE NAME(LNKLST00)
LNKLST ADD NAME(LNKLST00) DSN(SYS1.SERBLINK)
LNKLST ADD NAME(LNKLST00) DSN(NETV130.SCNMLNK1)  VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(NETV130.SEKGMOD1)  VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(NETV130.SEKGMOD2)  VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(IGY210.SIGYCOMP)   VOLUME(Z1RES2)
LNKLST ADD NAME(LNKLST00) DSN(FAN130.SEAGALT)    VOLUME(Z1RES2)
    ..
    ..
LNKLST ADD NAME(LNKLST00) DSN(your.load.pds)  ← Addition
LNKLST ACTIVATE NAME(LNKLST00)
```

To authorize the LOCO library, the current authorized program list, usually found in the PROGxx member of SYS1.PARMLIB, must be modified. The following example shows a partial listing of the APF portion of a PROGxx member before the LOCO library has been added. The exact format of the APF card should be verified in the Initialization and Tuning Guide for the MVS version that is executing in your installation.

```
APF FORMAT(DYNAMIC)
APF ADD DSNAME(SYS1.SERBLINK)    VOLUME(Z1RES2)
APF ADD DSNAME(SYS1.CSSLIB)      VOLUME(Z1RES1)
APF ADD DSNAME(IGY210.SIGYCOMP)  VOLUME(Z1RES2)
APF ADD DSNAME(FAN130.SEAGALT)   VOLUME(Z1RES2)
    ..
    ..
APF ADD DSNAME(DIT130.SDITMOD1)  VOLUME(Z1RES1)
```

The following example is a partial listing of the same PROGxx member with the LOCO library added.

```
APF FORMAT(DYNAMIC)
APF ADD DSNAME(SYS1.SERBLINK)    VOLUME(Z1RES2)
APF ADD DSNAME(SYS1.CSSLIB)      VOLUME(Z1RES1)
APF ADD DSNAME(IGY210.SIGYCOMP)  VOLUME(Z1RES2)
    ..
    ..
APF ADD DSNAME(DIT130.SDITMOD1)  VOLUME(Z1RES1)
APF ADD DSNAME(your.load.pds)    SMS  ← Addition
```

Method 2 Installation of the LOCO authorization SVC.

The second method of providing authorization for LOCO is by installing a LOCO supplied authorization SVC. If you have a previous version of LOCO currently installed, this step may be bypassed because the SVC has not been changed and does not require refreshing. The SVC is shipped in you.load.pds as member LOCOSVC. The LOCOSVC member must be copied to a library that is cataloged in the MVS master catalog. Usually this is a library that has the high level qualifier of SYS1. The following JCL will copy the LOCO SVC from the distribution library to a PDS called SYS1.LOCO.LPALIB.

```
//LOCOINST     JOB ..............................
//LOCOCOPY     EXEC PGM=IEBCOPY
//SYSPRINT     DD   SYSOUT=*
//INPUT        DD   DSN=your.load.pds,DISP=SHR
//OUTPUT       DD   DSN=SYS1.LOCO.LPALIB,
//                  DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(TRK,(5,1,1),RLSE)
//SYSIN       DD   *
 COPY I=INPUT,O=OUTPUT
 SELECT M=(LOCOSVC)
 /*
```

The SVC must be defined as a type 3 or 4 SVC. This can be done through modification of two SYS1.PARMLIB members. The IEASVCxx member defines the current user SVCs found in your operating environment. The following is an example of an IEASVCxx member where several user SVCs have been defined.

```
SVCPARM 255,REPLACE,TYPE(2)                           /*IMS  4.1*/
SVCPARM 254,REPLACE,TYPE(4)                           /*IMS  4.1*/
SVCPARM 216,REPLACE,TYPE(3),APF(NO),EPNAME(DFHCSVC)   /*CICS 6.2*/
SVCPARM 215,REPLACE,TYPE(6),APF(NO),EPNAME(DFHHPSVC)  /*CICS 6.2*/
SVCPARM 205,REPLACE,TYPE(2),APF(NO),EPNAME(IGC205)    /*IMS  6.1*/
SVCPARM 204,REPLACE,TYPE(4),APF(NO)                   /*IMS  6.1*/
SVCPARM 203,REPLACE,TYPE(2),APF(NO),EPNAME(IGC203)    /*IMS  5.1*/
SVCPARM 202,REPLACE,TYPE(4),APF(NO)                   /*IMS  5.1*/
```

Below, the example has been modified to add the LOCO SVC. The SVC number that was selected for the LOCO SVC is 252.

```
SVCPARM 255,REPLACE,TYPE(2)                           /*IMS  4.1*/
SVCPARM 254,REPLACE,TYPE(4)                           /*IMS  4.1*/
SVCPARM 252,REPLACE,TYPE(3),APF(NO),EPNAME(LOCOSVC)   /*LOCO ATH*/
SVCPARM 216,REPLACE,TYPE(3),APF(NO),EPNAME(DFHCSVC)   /*CICS 6.2*/
SVCPARM 215,REPLACE,TYPE(6),APF(NO),EPNAME(DFHHPSVC)  /*CICS 6.2*/
SVCPARM 205,REPLACE,TYPE(2),APF(NO),EPNAME(IGC205)    /*IMS  6.1*/
SVCPARM 204,REPLACE,TYPE(4),APF(NO)                   /*IMS  6.1*/
SVCPARM 203,REPLACE,TYPE(2),APF(NO),EPNAME(IGC203)    /*IMS  5.1*/
SVCPARM 202,REPLACE,TYPE(4),APF(NO)                   /*IMS  5.1*/
```

The LOCOSVC load module must be placed in the MVS LPA to act as an SVC. This can be done my making modifications to the IEALPAxx member in SYS1.PARMLIB that is currently in use at your installation. The following is an example of an IEALPAxx member before the LOCOSVC definition has been added.

```
INCLUDE LIBRARY(SYS1.CICSTS23.CICS.SDFHLPA)
        MODULES(DFHCSVC,
               DFHDSPEX,
               DFH99SVC,
               DFHDUMPX,
               DFHIRP
               )
```

The following is an example of the same IEALPAxx member after additions have been made for the LOCO SVC. Note that the library that contains the SVC must be a library that is cataloged in the MVS master catalog.

```
INCLUDE LIBRARY(SYS1.LOCO.LPALIB)  ← Addition
        MODULES(LOCOSVC,           ← Addition
                )                  ← Addition
INCLUDE LIBRARY(SYS1.CICSTS23.CICS.SDFHLPA)
        MODULES(DFHCSVC,
               DFHDSPEX,
               DFH99SVC,
               DFHDUMPX,
               DFHIRP
               )
```

An IPL of the MVS system with a CLPA option is now necessary to cause the LOCOSVC module to be placed in the MVS LPA.

Step 4. Make the LOCOINIT program available at IPL.

If you chose to authorize the LOCO Library via inclusion in the MVS link list you should skip this step. If other authorization methods were selected it is necessary to copy the LOCOINIT program to a link list library. The actual library is site dependent. For purposes of this example the library is assumed to be SYS1.LINKLIB. The following example JCL will copy the LOCOINIT program into the link list library SYS1.LINKLIB.

```
//LOCOINST     JOB ..............................
//LOCOCOPY     EXEC PGM=IEBCOPY
//SYSPRINT     DD   SYSOUT=*
//INPUT        DD   DSN=your.load.pds,DISP=SHR
//OUTPUT       DD   DSN=SYS1.LINKLIB,DISP=SHR
//SYSIN        DD   *
 COPY I=INPUT,O=OUTPUT
 SELECT M=(LOCOINIT)
 SELECT M=(LOCOSSSM)
/*
```

Step 5. Define LOCO as a subsystem to your MVS system.

You may pick any valid subsystem name that meets your installations naming conventions. The subsystems must be defined in the IEFSSNxx member of SYS1.PARMLIB using the selected subsystem name, LOCOINIT as the initialization program and the LOCO SVC number as the parm. If you do not use the LOCO SVC then the parm is not required. See the appropriate MVS level of the IBM manual “Initialization and Tuning Guide” for details on defining subsystems in the IEFSSNxx member.
The following is an example of an IEFSSNxx member before the addition of the LOCO definition.

```
SUBSYS SUBNAME(SMS)INITRTN(IGDSSIIN)
       INITPARM('ID=00,PROMPT=DISPLAY')
SUBSYS SUBNAME(JES2) PRIMARY(YES)  START(NO)
SUBSYS SUBNAME(RACF) INITRTN(IRRSSI00) INITPARM('#')
SUBSYS SUBNAME(TNF)
SUBSYS SUBNAME(LOGR) INITRTN(IXGSSINT)
SUBSYS SUBNAME(RRS)
SUBSYS SUBNAME(CICS)
SUBSYS SUBNAME(IRLM)
SUBSYS SUBNAME(JRLM)
```

The following is an example of an IEFSSNxx member after the addition of the LOCO subsystem definition for a subsystem named LOCO.

```
SUBSYS SUBNAME(SMS)INITRTN(IGDSSIIN)
       INITPARM('ID=00,PROMPT=DISPLAY')
SUBSYS SUBNAME(JES2) PRIMARY(YES)  START(NO)
SUBSYS SUBNAME(RACF) INITRTN(IRRSSI00) INITPARM('#')
SUBSYS SUBNAME(TNF)
SUBSYS SUBNAME(LOGR) INITRTN(IXGSSINT)
SUBSYS SUBNAME(RRS)
SUBSYS SUBNAME(CICS)
SUBSYS SUBNAME(IRLM)
SUBSYS SUBNAME(JRLM)
SUBSYS SUBNAME(LOCO) INITRTN(LOCOINIT) INITPARM='252' ← Addition
```

Either an IPL or a dynamic subsystem addition via the MVS console is required to inform the MVS system of the new subsystem name. LOCO may be installed before or after the IPL, but LOCOMAIN will execute only after the subsystem has been defined. The LOCOINIT program must be available via the MVS link list when the SETSSI command or IPL is perform or LOCO will not initialize.

To dynamically add the LOCO subsystem enter the following command from an MVS console. The INITPARM can be eliminated on the SETSSI command if the LOCO SVC is not being used for authorization.

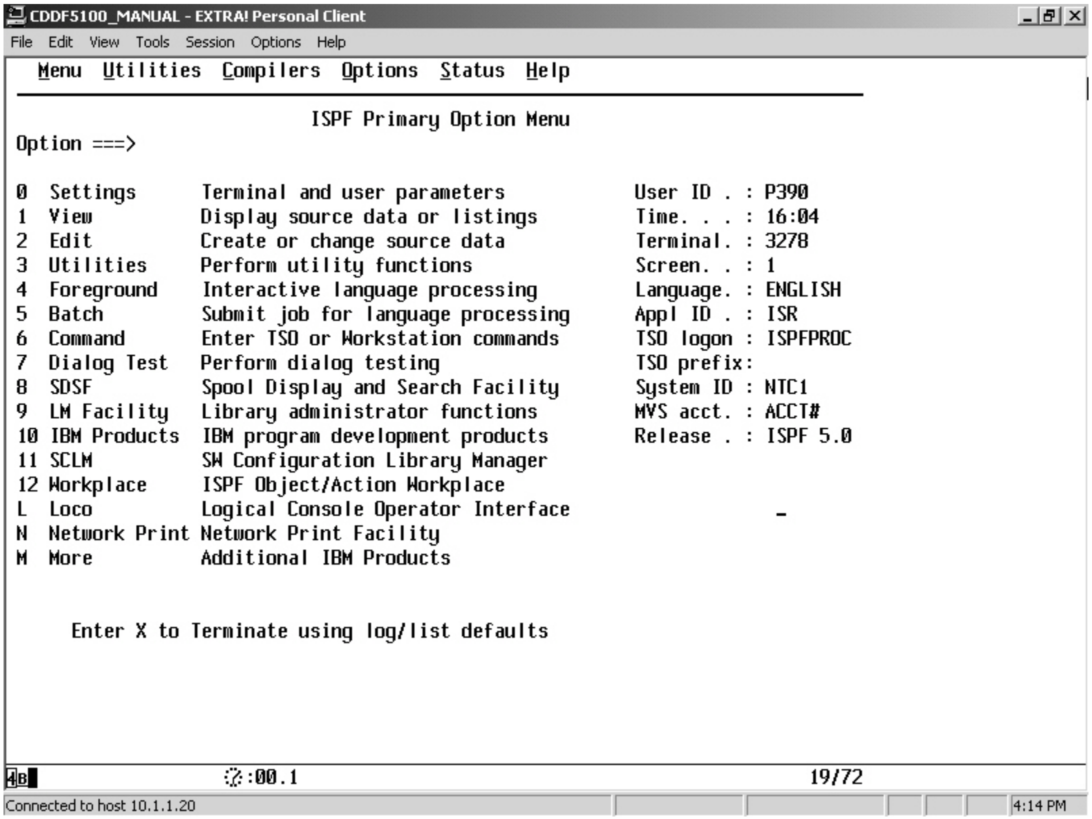
```
SETSSI ADD,SUBNAME=xxxx,INITRTN=LOCOINIT,INITPARM=yyy

Where xxxx = LOCO Subsystem name
      yyy  = The LOCO SVC number for authorization
```

Step 6. Install the LOCO TSO Interface

a. Modify the TSO procedure to include the LOCO TSO Interface Program libraries. The following libraries should be concatenated with the libraries found with the corresponding DDNAMEs. Exercise caution when modifying the TSO procedure. If a mistake is made that will cause a JCL error a TSO user will not be able to logon using the modified procedure.

DDNAME	Library
STEPLIB	your.load.pds
ISPPLIB	your.ispf.ispplib
ISPMLIB	your.ispf.ispmlib

To determine the TSO procedure that a TSO user is currently using, enter the primary option menu of ISPF. The screen should look similar to the following example. On the right side of the screen is a field labeled TSO logon. This field contains the TSO procedure name that is most likely the one to be modified. In the following example the TSO procedure name is ISPFPROC.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
  Menu  Utilities  Compilers  Options  Status  Help
 ------------------------------------------------------------------------------
                          ISPF Primary Option Menu
 Option ===>

 0  Settings      Terminal and user parameters          User ID . : P390
 1  View          Display source data or listings       Time. . . : 16:04
 2  Edit          Create or change source data          Terminal. : 3278
 3  Utilities     Perform utility functions             Screen. . : 1
 4  Foreground    Interactive language processing       Language. : ENGLISH
 5  Batch         Submit job for language processing    Appl ID . : ISR
 6  Command       Enter TSO or Workstation commands     TSO logon : ISPFPROC
 7  Dialog Test   Perform dialog testing                TSO prefix:
 8  SDSF          Spool Display and Search Facility     System ID : NTC1
 9  LM Facility   Library administrator functions       MVS acct. : ACCT#
 10 IBM Products  IBM program development products      Release . : ISPF 5.0
 11 SCLM          SW Configuration Library Manager
 12 Workplace     ISPF Object/Action Workplace
 L  Loco          Logical Console Operator Interface              _
 N  Network Print Network Print Facility
 M  More          Additional IBM Products


      Enter X to Terminate using log/list defaults




4B                :00.1                                  19/72
Connected to host 10.1.1.20                                           4:14 PM
```

The following is a partial listing of the ISPFPROC from the previous example before it has been modified to include the LOCO libraries. The changes have been indicated below. The double dots (..) indicate missing lines of information that have been removed to shorten the listing.

```
//*******************************************************************
//*
//*                 ISPF FULL-FUNCTION LOGON PROC
//*
//*******************************************************************
//ISPFPROC EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=175,
//             PARM='%ISPFCL'
//SYSUADS  DD  DISP=SHR,DSN=SYS1.UADS
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DISP=SHR,DSN=SYS1.LOCAL.CLIST
//         DD  DISP=SHR,DSN=ISP.SISPCLIB
           ..
           ..
//SYSEXEC  DD  DISP=SHR,DSN=ISP.SISPEXEC
//         DD  DISP=SHR,DSN=SOMMVS.SGOSREXX
           ..
//SYSHELP  DD  DISP=SHR,DSN=SYS1.HELP
//         DD  DISP=SHR,DSN=ISP.SISPHELP
           ..
           ..
//ISPMLIB  DD  DISP=SHR,DSN=ISP.SISPMENU
//         DD  DISP=SHR,DSN=SYS1.DFQMLIB
           ..
           ..
//ISPEXEC  DD  DISP=SHR,DSN=ISP.SISPEXEC
//         DD  DISP=SHR,DSN=SYS1.SBPXEXEC
//         ..
//         ..
//ISPLLIB  DD  DISP=SHR,DSN=GDDM.SADMMOD
           ..
           ..
//ISPPLIB  DD  DISP=SHR,DSN=SYS1.LOCAL.ISPFPNLS
//         DD  DISP=SHR,DSN=ISP.SISPPENU
//         DD  DISP=SHR,DSN=SYS1.DFQPLIB
           ..
           ..
//ISPSLIB  DD  DISP=SHR,DSN=ISP.SISPSLIB,UNIT=3390,VOL=SER=Z1RES1
//         DD  DISP=SHR,DSN=GIM.SGIMSENU,UNIT=3390,VOL=SER=Z1RES1
           ..
           ..
//ISPTLIB  DD  DISP=SHR,DSN=ISP.SISPTENU,UNIT=3390,VOL=SER=Z1RES1
//         DD  DISP=SHR,DSN=SYS1.DGTTLIB,UNIT=3390,VOL=SER=Z1RES1
           ..
           ..
           ..
```

The following is a partial listing of the ISPFPROC from the previous example after it has been modified to include the LOCO libraries. The changes have been indicated below. The double dots (..) indicate missing lines of information that have been removed to shorten the listing.

```
//*******************************************************************
//*
//*                 ISPF FULL-FUNCTION LOGON PROC
//*
//*******************************************************************
//ISPFPROC EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=175,
//             PARM='%ISPFCL'
//STEPLIB   DD  DISP=SHR,DSN=your.load.pds ← Addition of LOCO STEPLIB
//SYSUADS   DD  DISP=SHR,DSN=SYS1.UADS
//SYSLBC    DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC   DD  DISP=SHR,DSN=SYS1.LOCAL.CLIST
//          DD  DISP=SHR,DSN=ISP.SISPCLIB
           ..
           ..
//SYSEXEC   DD  DISP=SHR,DSN=ISP.SISPEXEC
//          DD  DISP=SHR,DSN=SOMMVS.SGOSREXX
           ..
//SYSHELP   DD  DISP=SHR,DSN=SYS1.HELP
//          DD  DISP=SHR,DSN=ISP.SISPHELP
           ..
           ..
//ISPMLIB   DD  DISP=SHR,DSN=your.ispf.ispmlib ← Addition of ISPMLIB
//          DD  DISP=SHR,DSN=ISP.SISPMENU
//          DD  DISP=SHR,DSN=SYS1.DFQMLIB
           ..
           ..
//ISPEXEC   DD  DISP=SHR,DSN=ISP.SISPEXEC
//          DD  DISP=SHR,DSN=SYS1.SBPXEXEC
//          ..
//          ..
//ISPLLIB   DD  DISP=SHR,DSN=GDDM.SADMMOD
           ..
           ..
//ISPPLIB   DD  DISP=SHR,DSN=you.ispf.ispplib ← Addition of ISPPLIB
//          DD  DISP=SHR,DSN=SYS1.LOCAL.ISPFPNLS
//          DD  DISP=SHR,DSN=ISP.SISPPENU
//          DD  DISP=SHR,DSN=SYS1.DFQPLIB
           ..
           ..
//ISPSLIB   DD  DISP=SHR,DSN=ISP.SISPSLIB,UNIT=3390,VOL=SER=Z1RES1
//          DD  DISP=SHR,DSN=GIM.SGIMSENU,UNIT=3390,VOL=SER=Z1RES1
           ..
           ..
//ISPTLIB   DD  DISP=SHR,DSN=ISP.SISPTENU,UNIT=3390,VOL=SER=Z1RES1
//          DD  DISP=SHR,DSN=SYS1.DGTTLIB,UNIT=3390,VOL=SER=Z1RES1
           ..
           ..
           ..
```

b. Modify your ISPF primary option panel or other appropriate panel to include the LOCO TSO Interface Program. The following panel definition should be executed as a result of being selected. An example modified primary option panel is found in your.ispf.panel.library(#PRIMOPT)

To locate your primary option panel, examine the TSO procedure and find the ISPPLIB DDNAME. The ISR@PRIM member (default name for the primary option member) will be found in one of the first concatenations of libraries for this DDNAME. The following is a partial listing of the default ISPF primary option panel furnished my IBM (before modification). The panel you find may be slightly different that this example. The changes have been indicated below. The double dots (..) indicate missing lines of information that have been removed to shorten the listing.

```
 )PANEL KEYLIST(ISRSAB,ISR) IMAGE(&ZIMGNAM,&ZIMGROW,&ZIMGCOL)
 )ATTR DEFAULT(   ) FORMAT(MIX)      /* ISR@PRIM - ENGLISH - 5.0 */
  0B TYPE(AB)
  0D TYPE(PS)
  04 TYPE(ABSL) GE(ON)
    ..
    ..
 )AREA SAREA39
 0  Settings      Terminal and user parameters
 1  View          Display source data or listings
 2  Edit          Create or change source data
 3  Utilities     Perform utility functions
 4  Foreground    Interactive language processing
 5  Batch         Submit job for language processing
 6  Command       Enter TSO or Workstation commands
 7  Dialog Test   Perform dialog testing
 8  LM Facility   Library administrator functions
 9  IBM Products  IBM program development products
 10 SCLM          SW Configuration Library Manager
 11 Workplace     ISPF Object/Action Workplace
)INIT
     ..
     ..
)PROC
IF (&ZCSTF='1') .ATTR(GRPBOX1) = 'WIDTH(0)'
IF (.CURSOR = TMPROWS AND &ZCMD = ' ')
&ZSAR  =TRANS(&ZSCREEN 1,&ZSAREA1 2,&ZSAREA2 3,&ZSAREA3 4,&ZSAREA4
 IF (&ZSAR = 'CAL','UPS','SES') &ZCMD = 'SP'
&ZSEL = TRANS (TRUNC (&ZCMD,'.')
  0,'PGM(ISPISM) SCRNAME(SETTINGS)'
  1,'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW)'
  2,'PGM(ISREDIT) PARM(P,ISREDM01) SCRNAME(EDIT)'
  3,'PANEL(ISRUTIL) SCRNAME(UTIL)'
  4,'PANEL(ISRFPA) SCRNAME(FOREGRND)'
  5,'PGM(ISRJB1) PARM(ISRJPA) SCRNAME(BATCH) NOCHECK'
  6,'PGM(ISRPTC) SCRNAME(CMD)'
  7,'PGM(ISPYXDR) PARM(&ZTAPPLID) SCRNAME(DTEST) NOCHECK'
  8,'PANEL(ISRLPRIM) SCRNAME(LMF)'
  9,'PANEL(ISRDIIS) ADDPOP'
 10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
 11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
  X,EXIT
SP,'PGM(ISPSAM) PARM(PNS)'
 ' ',' '
   *,'?' )
&ZTRAIL=.TRAIL
)PNTS
     ..
     ..
)END
```

The following is a partial listing of the modified ISPF primary option. The changes have been indicated below. The double dots (..) indicate missing lines of information that have been removed to shorten the listing.

```
 )PANEL KEYLIST(ISRSAB,ISR) IMAGE(&ZIMGNAM,&ZIMGROW,&ZIMGCOL)
 )ATTR DEFAULT(   ) FORMAT(MIX)      /* ISR@PRIM - ENGLISH - 5.0 */
  0B TYPE(AB)
  0D TYPE(PS)
  04 TYPE(ABSL) GE(ON)
    ..
    ..
 )AREA SAREA39
 0  Settings      Terminal and user parameters
 1  View          Display source data or listings
 2  Edit          Create or change source data
 3  Utilities     Perform utility functions
 4  Foreground    Interactive language processing
 5  Batch         Submit job for language processing
 6  Command       Enter TSO or Workstation commands
 7  Dialog Test   Perform dialog testing
 8  LM Facility   Library administrator functions
 9  IBM Products  IBM program development products
 10 SCLM          SW Configuration Library Manager
 11 Workplace     ISPF Object/Action Workplace
 L  Loco          Logical Console Operator Interface  ←  Addition
)INIT
     ..
     ..
)PROC
IF (&ZCSTF='1') .ATTR(GRPBOX1) = 'WIDTH(0)'
IF (.CURSOR = TMPROWS AND &ZCMD = ' ')
&ZSAR  =TRANS(&ZSCREEN 1,&ZSAREA1 2,&ZSAREA2 3,&ZSAREA3 4,&ZSAREA4
 IF (&ZSAR = 'CAL','UPS','SES') &ZCMD = 'SP'
&ZSEL = TRANS (TRUNC (&ZCMD,'.')
  0,'PGM(ISPISM) SCRNAME(SETTINGS)'
  1,'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW)'
  2,'PGM(ISREDIT) PARM(P,ISREDM01) SCRNAME(EDIT)'
  3,'PANEL(ISRUTIL) SCRNAME(UTIL)'
  4,'PANEL(ISRFPA) SCRNAME(FOREGRND)'
  5,'PGM(ISRJB1) PARM(ISRJPA) SCRNAME(BATCH) NOCHECK'
  6,'PGM(ISRPTC) SCRNAME(CMD)'
  7,'PGM(ISPYXDR) PARM(&ZTAPPLID) SCRNAME(DTEST) NOCHECK'
  8,'PANEL(ISRLPRIM) SCRNAME(LMF)'
  9,'PANEL(ISRDIIS) ADDPOP'
 10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
 11,'PGM(ISRUDA) PARM(ISRWORK) SCRNAME(WORK)'
  L,'PANEL(LOCOTSM1) NEWAPPL(LOCO) PASSLIB'  ← Addition
  X,EXIT
SP,'PGM(ISPSAM) PARM(PNS)'
 ' ',' '
   *,'?' )
&ZTRAIL=.TRAIL
)PNTS
     ..
)END
```

c. Modify the IKJTSOxx SYS1.PARMLIB member to add the LOCO subsystem interface authorized program LOCOTSPZ. LOCOTSPZ is called through the TSO Service Facility and needs to be added to the AUTHTSF list of programs. The following is an example of this addition.

```
AUTHTSF NAMES(           /* PROGRAMS TO BE AUTHORIZED */ +
                         /* WHEN CALLED THROUGH THE   */ +
                         /* TSO SERVICE FACILITY.     */ +
    LOCOTSPZ             /* LOCO TSO INTERFACE PGM    */ + ← Addition
    CSFDAUTH             /* ICSF                      */ +
    IEBCOPY              /*                           */ +
    ICQASLI0             /*                           */ +
    IKJEFF76             /*                           */   )
```

Step 7. Define security for the LOCO TSO Interface

The LOCO TSO interface utilizes an external security manager such as RACF, ACF2, or Top Secret to verify a user’s functional authorization. SAF calls are made through out the interface to verify the user’s permission to execute the LOCO function that is being attempted. If the user is found to be unauthorized, access to the LOCO function is denied. Profile checking occurs at two levels. First the users ability to access the LOCO subsystem interface is determined. Second the users ability to execute the requested function within the specified LOCO subsystem is checked. If at any time, the user’s authorization fails, access to the requested function or LOCO subsystem is denied. Through the use of generic profiles, the LOCO security scheme can be as simple or as complex as the installation requires. The security requirements for each individual LOCO TSO function is described as part of that function’s description found in the TSO Users Guide section of this manual. Details on how to define resource profiles can also be found in the TSO Users Guide in the Security for the TSO Interface section.

Security profiles must be defined before any user can access the LOCO TSO Interface.

Step 8. Print and review the sample LOCO control tables from the LOCO source PDS. The following JCL is member STEP6JCL in 'your.source.pds'. These tables can be used as examples when preparing LOCO control tables for use in your installation.

```
//LOCOPRNT   JOB   ..........................
//PRINT      PROC
//GENER      EXEC    PGM=IEBGENER
//SYSPRINT   DD      SYSOUT=*
//SYSUT2     DD      SYSOUT=*
//SYSUT1     DD      DSN=your.source.pds(&MEMBER),DISP=SHR
//SYSIN      DD      DUMMY
//                   PEND
//SAMP1      EXEC    PRINT,MEMBER=SAMPLE1
//SAMP2      EXEC    PRINT,MEMBER=SAMPLE2
//SAMP3      EXEC    PRINT,MEMBER=SAMPLE3
//SAMP4      EXEC    PRINT,MEMBER=SAMPLE4
//SAMP5      EXEC    PRINT,MEMBER=SAMPLE5
//SAMP6      EXEC    PRINT,MEMBER=SAMPLE6
//SAMP7      EXEC    PRINT,MEMBER=SAMPLE7
//SAMP8      EXEC    PRINT,MEMBER=SAMPLE8
//SAMP9      EXEC    PRINT,MEMBER=SAMPLE9
//SAMP10     EXEC    PRINT,MEMBER=SAMPLE10
```

Step 9. Define LOCO Control Table for use at your installation.

If this is a new installation of LOCO you will want to create and build a set of LOCO Control tables for your installation. Use the tables printed in step 7 as examples. The easiest way to accomplish this task is to use the standard ISPF editor to create source for a new LOCO table. After the initial LOCOBLDR execution, the table will be editable via the LOCO TSO interface.

If you are upgrading LOCO from a previous release it will be necessary to execute LOCOBLDR against all of your current tables.

Step 10. Optionally add LOCO to the installation’s MVS Automatic Restart Manager (ARM) policy.

LOCO is available for automation restart by the MVS Automatic Restart Manager. When LOCO initializes, LOCOMAIN registers with ARM. If your installation’s ARM policy specifies restart rules for LOCO it will be restarted in the event of a failure according to these rules. For additional information on ARM and the definition of ARM policies please refer to the IBM document, MVS Setting Up a Sysplex, GC28-1779.

LOCO utilizes the following ARM element names and element types.

ARM Element Type	**LOCO**
ARM Element Name	**LOCO_xxxx**

xxxx is the LOCO susbsytem name of the LOCO that is registering with ARM.

Getting Started

Analyzing SYSLOG to create LOCO Tables

The LOCOLOG job reads a copy of your SYSLOG data set and lists all WTO(R)s and REPLYs together. It sorts all WTO(R)s by message number to provide potential candidates for LOCO automatic action.

```
//ANALYSE   JOB    .....................
//LOG       EXEC   PGM=LOCOLOG,REGION=1024K,PARM=nnn
//STEPLIB   DD     DSN=your.loco.loadlib,DISP=SHR
//SORTLIB   DD     DSN=SYS1.SORTLIB,DISP=SHR
//SYSPRINT  DD     SYSOUT=*
//SYSOUT    DD     SYSOUT=*
//SORTMSG   DD     SYSOUT=*
//SORTWK01  DD     DSN=&SORTWK1,UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK02  DD     DSN=&SORTWK2,UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK03  DD     DSN=&SORTWK3,UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN     DD     DSN=your.copy.of.syslog,DISP=SHR,
//                 UNIT=sysda/tape,VOL=SER=vvvvvv,
//                 DCB=(RECFM=FB,LRECL=rrr,BLKSIZE=bbbb)
```

Notes:

1. Make a copy of your SYSLOG data and provide the appropriate JCL for SYSIN for LOCOLOG to read the SYSLOG data.

2. The PARM is a three digit number specifying the position (relative to zero) of the @ or * which precedes the WTOR text in the syslog records. The default is 24. For example:

```
          1         2         3         4         5
0....5....0....5....0....5....0....5....0....5....0
0000 22.01.01           *01 IFB010D ENTER 'IPL REASON ...
                        ^
                        |__This is position 24.
```

Initializing the LOCO Subsystem

LOCO is initialized at IPL time by the standard MVS subsystem mechanism controlled by the IEFSSNxx member of PARMLIB. The program module LOCOINIT is the LOCO initialization program that is invoked for each LOCO subsystem defined in IEFSSNxx.

Specifying the Operational Characteristics of LOCOMAIN

The operational characteristics of the execution of LOCOMAIN are controlled through the use of keywords and values specified in the PARM field on the LOCOMAIN execute statement. The following describe the possible parameters that can be specified.

Parameter	**Values**	**Description**
SSN=	xxxx	xxxx is the LOCO subsystem name to be used for this execution of LOCOMAIN. This is a required paramter.
SVC=	xxx	SVC number that was assign to the LOCO authorization SVC at installation time
DS=	Y/N	Y indicates the WTO blocks are to be held in a LOCOMAIN created dataspace. N indicates the WTO blocks are to be held in CSA. DS=N is the default.
MEMBER=	xxxxxxxx	xxxxxxxx is the member name of the LOCO table to load and execution at LOCOMAIN initialization time. This is a required parameter.
CAFC=	Y/N	Y indicates LOCOMAIN should include CAFC support at initialization time. N indicates that support will not be available during this execution of LOCOMAIN. CAFC=N is the default.
DBCT=	Y/N	Y indicates LOCOMAIN should include DB Control Batch Interface support at initialization time. N indicates that support will not be available during this execution of LOCOMAIN. DBCT=N is the default.
NOREISSUE		LOCO should ignore WTOs that have been reissued by XCF in a Sysplex environment. A reissued message is specified by z/OS via the WQERISS bit being on in the WQEFLG1 field of the WQE.
TRACE		The presence of the TRACE keyword in the PARM field will cause LOCO to produce an internal trace of LOCO activities to a SYSOUT dataset. This option should only be activated at the request of NETEC technical support. ***This option will produce a high volume of output.*** The option can also be activated through the LOCO TSO interface after LOCOMAIN has begun execution.

Parameter	Values	Description
TRWQE		The presence of the TRWQE keyword in the PARM field will cause LOCO to produce an internal trace to a SYSOUT dataset of all WQEs that enter the LOCO subsystem SVC 35 exit. The TRACE option must also be specified to obtain any output from the TRWQE keyword. This option should only be activated at the request of NETEC technical support. ***This option will produce a high volume of output.*** The option can also be activated through the LOCO TSO interface after LOCOMAIN has begun execution.
TRMSG		The presence of the TRMSG keyword in the PARM field will cause LOCO to produce an internal trace to a SYSOUT dataset of all WTO blocks receive by LOCOMAIN from the LOCO subsystem SVC 35 exit. The TRACE option must also be specified to obtain any output from the TRMSG keyword. This option should only be activated at the request of NETEC technical support. ***This option will produce a high volume of output.*** The option can also be activated through the LOCO TSO interface after LOCOMAIN has begun execution.
REPORT		The presence of the REPORT keyword in the PARM field will cause LOCO to execute in report mode. When in report mode, LOCO will output to SYSPRINT the actions that would have been taken if LOCO were executing in real time. Report mode is useful when you want to check that LOCO tables are taking the actions that were intended. Report mode is executed as if LOCO were running between specific dates and times. These dates and times are specified through the LODATE and HIDATE parameters that are specified on the LOCOMAIN execution PARM.
DATE=	mm/dd/yyyy	This field or the LODATE= and HIDATE= field is required if the REPORT keyword is specified in the LOCOMAIN PARM field. This field specifies the date that LOCO is to use for report mode execution.
LODATE=	mm/dd/yyyy	This field is required if the REPORT keyword is specified in the LOCOMAIN PARM field. This field specifies the beginning date that LOCO is to use for report mode execution.

Parameter	Values	Description
HIDATE=	mm/dd/yyyy	This field is required if the REPORT keyword is specified in the LOCOMAIN PARM field. This field specifies the ending date that LOCO is to use for this report mode execution.
SIMULATE		The presence of the SIMULATE keyword in the LOCOMAIN PARM instructs LOCO to simulate all actions during this execution of LOCO. LOCOMAIN will record all actions that it would have normally taken during this execution but the actual commands and actions will not be issued or actions taken. Simulate mode is useful when you want to check that LOCO tables are taking the actions that were intended in the real execution environment.

Reporting and Simulating LOCO Actions

LOCO has the ability to execute either in a simulation or a reporting mode. These modes are specified to LOCOMAIN by specifying SIMULATE for simulation mode or REPORT for report mode in the LOCOMAIN PARM field. When LOCO is executing in report mode, LOCOMAIN will process the LOCO table specified by the MEMBER= parameter value as if it had been executed either on the DATE= parameter or from the LODATE= parameter to the HIDATE= parameter. The results of the execution are output to REPORT ddname. No real actions are taken during a report mode execution.
When LOCO is executing in simulate mode, LOCOMAIN will process the LOCO table specified by the MEMBER= parameter value in real time. The results of the execution are output to the SIMULATE ddname. No real actions are taken during a simulate mode execution. Both simulate and report modes of execution are useful when validating new tables before that are placed into production.

The following JCL will execute LOCO in report mode.

```
//LOCOMAIN   JOB    .................
//LOCO       EXEC PGM=LOCOMAIN,
//           DPRTY=(15,15),
//           PERFORM=9,
//           TIME=1440,
//           REGION=1024K,
// PARM='MEMBER=tabname,SSN=xxxx,REPORT,DATE=01/01/2004'
//* ---------------------------------------------------------------- *
//STEPLIB    DD     DSN=your.loco.loadlib,DISP=SHR
//           DD     DSN=your.CAFC.loadlib,DISP=SHR   ← Optional
//           DD     DSN=your.DBCT.loadlib,DISP=SHR   ← Optional
//TABLE      DD     DSN=your.loco.table.library,DISP=SHR
//SUBMIT     DD     DSN=your.loco.job.stream.library,DISP=SHR
//CRITICAL   DD     DSN=your.loco.critical.file,DISP=SHR,DCB=BUFNO=1
//CRITCAL1   DD     DSN=your.loco.critical.file1,DISP=SHR,DCB=BUFNO=1
//CRITCAL2   DD     DSN=your.loco.critical.file2,DISP=SHR,DCB=BUFNO=1
//CRITCAL3   DD     DSN=your.loco.critical.file3,DISP=SHR,DCB=BUFNO=1
//*
//INTRDR     DD     SYSOUT=(A,INTRDR),DCB=BLKSIZE=80
//SYSPRINT   DD     SYSOUT=*
//SIMULATE   DD     SYSOUT=*
//REPORT     DD     SYSOUT=*
//SYSABEND   DD     SYSOUT=*
//ALOGSUBT   DD     SYSOUT=*
//CAFCSUBT   DD     SYSOUT=*
//DBCTSUBT   DD     SYSOUT=*
//LOCOALOG   DD     SYSOUT=*
```

The following JCL will execute LOCO in simulate mode.

```
//LOCOMAIN   JOB    .................
//LOCO       EXEC PGM=LOCOMAIN,
//           DPRTY=(15,15),
//           PERFORM=9,
//           TIME=1440,
//           REGION=1024K,
// PARM='MEMBER=tabname,SSN=xxxx,SIMULATE'
//* ---------------------------------------------------------------- *
//STEPLIB    DD     DSN=your.loco.loadlib,DISP=SHR
//           DD     DSN=your.CAFC.loadlib,DISP=SHR    ← Optional
//           DD     DSN=your.DBCT.loadlib,DISP=SHR    ← Optional
//TABLE      DD     DSN=your.loco.table.library,DISP=SHR
//SUBMIT     DD     DSN=your.loco.job.stream.library,DISP=SHR
//CRITICAL   DD     DSN=your.loco.critical.file,DISP=SHR,DCB=BUFNO=1
//CRITCAL1   DD     DSN=your.loco.critical.file1,DISP=SHR,DCB=BUFNO=1
//CRITCAL2   DD     DSN=your.loco.critical.file2,DISP=SHR,DCB=BUFNO=1
//CRITCAL3   DD     DSN=your.loco.critical.file3,DISP=SHR,DCB=BUFNO=1
//*
//INTRDR     DD     SYSOUT=(A,INTRDR),DCB=BLKSIZE=80
//SYSPRINT   DD     SYSOUT=*
//SIMULATE   DD     SYSOUT=*
//REPORT     DD     SYSOUT=*
//SYSABEND   DD     SYSOUT=*
//ALOGSUBT   DD     SYSOUT=*
//CAFCSUBT   DD     SYSOUT=*
//DBCTSUBT   DD     SYSOUT=*
//LOCOALOG   DD     SYSOUT=*
```

Starting the LOCO Subsystem

This is the main LOCO job. It interrogates WTOs, WTORs and COMMANDS coming from the MVS system and performs actions as dictated by the current control table. If the user is going to utilize the CAFCCMD or DBCTCMD commands in a LOCO table, STEPLIBs must be added to the LOCOMAIN step that contain the DSNAMEs of the CICS Application File Control system and the DBControl Batch Interface.

```
//LOCOMAIN   JOB    .................
//LOCO       EXEC PGM=LOCOMAIN,
//           DPRTY=(15,15),
//           PERFORM=9,
//           TIME=1440,
//           REGION=1024K,
// PARM='MEMBER=tabname,SSN=xxxx,otherparms'
//* --------------------------------------------------------------- *
//STEPLIB    DD     DSN=your.loco.loadlib,DISP=SHR
//           DD     DSN=your.CAFC.loadlib,DISP=SHR   ← Optional
//           DD     DSN=your.DBCT.loadlib,DISP=SHR   ← Optional
//TABLE      DD     DSN=your.loco.table.library,DISP=SHR
//SUBMIT     DD     DSN=your.loco.job.stream.library,DISP=SHR
//CRITICAL   DD     DSN=your.loco.critical.file,DISP=SHR,DCB=BUFNO=1
//CRITCAL1   DD     DSN=your.loco.critical.file1,DISP=SHR,DCB=BUFNO=1
//CRITCAL2   DD     DSN=your.loco.critical.file2,DISP=SHR,DCB=BUFNO=1
//CRITCAL3   DD     DSN=your.loco.critical.file3,DISP=SHR,DCB=BUFNO=1
//*
//INTRDR     DD     SYSOUT=(A,INTRDR),DCB=BLKSIZE=80
//SYSPRINT   DD     SYSOUT=*
//SIMULATE   DD     SYSOUT=*
//REPORT     DD     SYSOUT=*
//SYSABEND   DD     SYSOUT=*
//ALOGSUBT   DD     SYSOUT=*
//CAFCSUBT   DD     SYSOUT=*
//DBCTSUBT   DD     SYSOUT=*
//LOCOALOG   DD     SYSOUT=*
```

NOTES:

1. The DPRTY and PERFORM keywords should be set to the same values you have on your JES proc. The effectiveness of LOCO is somewhat diminished if it runs at a low priority.
2. Tabname in the PARM is the initial LOCO control table for LOCOMAIN to use, and xxxx is the LOCO subsystem name you have chosen.
3. The STEPLIB statement is not necessary if LOCOMAIN resides in an APF authorized LNKLST library.

4. The TABLE statement defines the PDS containing the LOCO object control tables.
5. The SUBMIT statement defines the PDS containing job streams for submission by the @> SUBMIT command.
6. The SYSPRINT statement is used for tracing and debugging as directed by NETEC.
7. Your.loco.job.stream.library may be any source library containing job streams. LOCO explicitly provides a /*EOF card to terminate the job stream. The submit JCL may not specify a concatenation of data sets - only a single data set.
8. Your.loco.critical.file requires the following DCB paramters: RECFM=FA and BLKSIZE=121.
9. If SIMULATE is included in the EXEC PARM, LOCOMAIN will print all actions on the SIMULATE data set rather than issuing commands to MVS.
10. If REPORT is included in the EXEC PARM, LOCOMAIN will create a formatted report of the specified LOCO control table on the REPORT data set and will terminate. No ACTIONs from the LOCO control table will be executed. Either a date must be specified by DATE=mm/dd/yyyy or a date range must be specified by LODATE=mm/dd/yyyy and HIDATE=mm/dd/yyyy in the PARM. LOCOMAIN will report the time initiated actions as they would occur for the given date range along with all other actions and filter conditions.
11. If NOREISSUE is included in the EXEC PARM, LOCOMAIN will ignore WTOs and WTORs which are reissued by XCF in a sysplex environment. A count of ignored messages can be displayed by issuing the @STATUS command.

The following proc can be used to start LOCOMAIN from the MVS console:

```
//LOCOX      PROC M=DUMMY
//IEFPROC    EXEC   PGM=LOCOMAIN,DPRTY=(15,15),PEFFORM=9,
//           PARM='MEMBER=&M,SSN=&S'
//STEPLIB    DD     DSN=your.loco.loadlib,DISP=SHR
//TABLE      DD     DSN=your.loco.table.library,DISP=SHR
//SUBMIT     DD     DSN=your.loco.job.stream.library,DISP=SHR
//CRITICAL   DD     DSN=your.loco.critical.file,DISP=SHR,DCB=BUFNO=1
//CRITCAL1   DD     DSN=your.loco.critical.file1,DISP=SHR,DCB=BUFNO=1
//CRITCAL2   DD     DSN=your.loco.critical.file2,DISP=SHR,DCB=BUFNO=1
//CRITCAL3   DD     DSN=your.loco.critical.file3,DISP=SHR,DCB=BUFNO=1
//INTRDR     DD     SYSOUT=(A,INTRDR),DCB=BLKSIZE=80
//SYSPRINT   DD     SYSOUT=*
//SYSABEND   DD     SYSOUT=*
```

```
Example START command:   S LOCOX,M=tabname,S=LCOJ
```

Utility Guide

Creating and Maintaining LOCO Tables

Creating LOCO Tables with LOCOBLDR

The LOCOBLDR job compiles source control tables in batch into object tables to be used by LOCOMAIN. The LOCOBLDR process is automatically invoked when a LOCO tables has been modified through the LOCO TSO/ISPF interface.

```
//COMPILE    JOB     ..........................
//BUILDER    EXEC    PGM=LOCOBLDR,REGION=1024K
//STEPLIB    DD      DSN=your.loco.loadlib,DISP=SHR
//SORTLIB    DD      DSN=SYS1.SORTLIB,DISP=SHR
//SYSPRINT   DD      SYSOUT=*
//SYSOUT     DD      SYSOUT=*
//SORTIN     DD      DSN=&SORTIN,UNIT=SYSDA,SPACE=(CYL,(4,1))
//SORTOUT    DD      DSN=&SORTOUT,UNIT=SYSDA,SPACE=(CYL,(4,1)),
//                   DCB=(RECFM=FB,LRECL=150,BLKSIZE=6000)
//TABLE      DD      DSN=your.loco.table.library(tabname),DISP=SHR
//SYSIN      DD      *
LOCO source control table
/*
```

Notes:

1. The SORTLIB statement is not necessary if SORTLIB is in your LNKLST.
2. The STEPLIB statement is not necessary if you have LINKed LOCOBLDR into a LNKLST library.
3. The SYSOUT statement is for SORT messages and can be changed to DD DUMMY.
4. The SYSIN statement is for the input source control table; it may be specified as a PDS and member name. LOCOBLDR expects 80 byte records blocked as you wish.
5. The TABLE statement is the output object control table in 150 byte records blocked up to 20k bytes. LOCOMAIN uses this same data set as input and has it OPEN only during actual table loading (initialization and @LOAD command). If you specify DISP=SHR, then you may compile LOCO control tables while LOCOMAIN is running.

Creating LOCO Tables with the LOCO TSO/ISPF Interface

LOCO tables can be created through the LOCO TSO/ISPF interface by issuing the edit command (E) with a member name that does not exist. The user can them enter the LOCO table source into ISPF until the table is complete. At that time, when the PF3 key is pressed, the table will be passed through the LOCOBLDR process and saved in the table library.

Managing LOCO Table Source

LOCOIMEX is a batch utility program to provide users with the ability to import and export LOCO table source into and out of the LOCO table library.

The following is an example of the basic JCL necessary to execute the LOCO table import/export utility. If the INDD, OUTDD, or TABLEDD keywords are utilized on the IMPORT and EXPORT commands, DD statements must be added to the JCL stream for these data sets.

```
//LOCOIMEX   JOB    .................
//STEP1      EXEC PGM=LOCOIMEX,
//           REGION=1024K,
//           PARM='TRACE'
//* -------------------------------------------------------------------
//STEPLIB    DD     DSN=your.loco.loadlib,DISP=SHR
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//SYSLIST    DD     SYSOUT=*
//SYSIN      DD     *
          ……  IMPORT/EXPORT CONTROL CARDS HERE ……
/*
```

IMPORT Command

The IMPORT command is utilized to move LOCO table source from a sequential or partitioned data set into the LOCO table library. Optionally, the table source may be listed to the SYSLIST ddname and may be processed through the LOCOBLDR process before being placed in the LOCO table library. If the source is not processed by LOCOBLDR, it will be placed into the LOCO table library and may be edited via the LOCO table maintenance process under TSO/ISPF but it cannot be loaded my LOCOMAIN until the source has been successfully processed by LOCOBLDR. The format of the IMPORT command follows.

COMMAND	OPERANDS
IMPORT	INDD(dddddddd) - INDSN(a.b.c.d) - TABLEDD(tttttttt) - TABLEDSN(e.f.g.h) - LIST(YES/NO) - BUILD(YES/NO) - MEMBER(mmmmmmmm,nnnnnnnn,REPLACE)

INDD dddddddd is the DDNAME in the LOCOIMEX JCL that is to be used as input for the import process. The INDD can either be a sequential or partitioned data set. The INDD keyword is mutually exclusive with the INDSN keyword.

INDSN a.b.c.d is the DSNAME of the data set to be used as input for the import process. The INDSN can be either a sequential or partitioned data set. The INDSN keyword is mutually exclusive with the INDD keyword.

TABLEDD tttttttt is the DDNAME in the LOCOIMEX JCL that is the LOCO table library that is the target for the source that is to be imported. The TABLEDD must be a PDS that contains LOCO tables. The TABLEDD keyword is mutually exclusive with the TABLEDSN keyword.

TABLEDSN e.f.g.h is the DSNAME of the LOCO table library that is the target for the source that is to be imported. The TABLEDSN must be a PDS that contains LOCO tables. The TABLEDSN keyword is mutually exclusive with the TABLEDD keyword.

LIST YES/NO indicates that the source is to be listed on the SYSLIST data set (when LIST(YES) is specified). The default is NO.

BUILD YES/NO indicates that the exported output should contain ./ ADD or ./ REPL control cards between members. If the output data set is a sequential data set the ./ ADD or ./ REPL control cards will separate each member with a ./ ENDUP control card as the last member. If the output is a PDS each member of the PDS will start with a ./ ADD or ./ REPL control card and end with a ./ ENDUP. The default is not include IEBUPDTE format control cards.

MEMBER mmmmmmmm indicates the member name that is to be imported. The member keyword is only valid when the INDD or INDSN is a PDS. The member name may terminate with an asterisk (*) indicating that it is a generic name. If the member name is specified as a single asterisk (*) all members of the input source library PDS will be imported. If nnnnnnnn is specified and mmmmmmmm is not a generic name, member mmmmmmmm will be renamed to member nnnnnnnn in the output table library. If REPLACE is not specified, LOCO tables members that already exist in the LOCO table library specified by TABLEDD or TABLEDSN will not be replaced with identical named members from the input source PDS.

EXPORT Command

The EXPORT command is utilized to move LOCO table source from the LOCO table library to a sequential or partitioned data set. Optionally, the table source may be listed to the SYSLIST DDNAME and may have ./ ADD or ./ REPL cards added to the output stream. The format of the EXPORT command follows.

COMMAND	OPERANDS
EXPORT	OUTDD(dddddddd) - OUTDSN(a.b.c.d) - TABLEDD(tttttttt) - TABLEDSN(e.f.g.h) - LIST(YES/NO) - FORMAT(IEBUPDTE) - MEMBER(mmmmmmmm,nnnnnnnn,REPLACE)

OUTDD dddddddd is the DDNAME in the LOCOIMEX JCL that is to be the target of the exported table source. The OUTDD can either be a sequential or partitioned data set. The OUTDD keyword is mutually exclusive with the OUTDSN keyword.

OUTDSN a.b.c.d is the DSNAME of the target data set for the exported table source. The OUTDSN can be either a sequential or partitioned data set. The OUTDSN keyword is mutually exclusive with the OUTDD keyword.

TABLEDD tttttttt is the DDNAME in the LOCOIMEX JCL that is to be the LOCO table library that contains the source to be exported. The TABLEDD must be a PDS that contains LOCO tables. The TABLEDD keyword is mutually exclusive with the TABLEDSN keyword.

TABLEDSN e.f.g.h is the DSNAME of the LOCO table library that contains the source to be exported. The TABLEDSN must be a PDS that contains LOCO tables. The TABLEDSN keyword is mutually exclusive with the TABLEDD keyword.

LIST YES/NO indicates that the source is to be listed on the SYSLIST data set (when LIST(YES) is specified). The default is NO.

FORMAT IEBUPDTE indicates that the exported output should contain ./ ADD or ./ REPL control cards between members. If the output data set is a sequential data set the ./ ADD or ./ REPL control cards will separate each member with a ./ ENDUP control card as the last member. If the output is a PDS each member of the PDS will start with a ./ ADD or ./ REPL control card and end with a ./ ENDUP. The default is not include IEBUPDTE format control cards.

MEMBER mmmmmmmm indicates the member name that is to be exported. The member name may terminate with an asterisk (*) indicating that it is a generic name. If the member name is specified as a single asterisk (*) all members of the LOCO table library will be exported. If the output data set is a sequential file and REPLACE is specified and FORMAT(IEBUPDTE) is specified, ./ REPL cards will be placed in the exported source instead of ./ ADD control cards. If the output is a PDS and REPLACE is not specified, members that already exist in the output data set with identical names will not be replaced. If nnnnnnnn is specified and mmmmmmmm is not a generic name, member mmmmmmmm will be renamed to member nnnnnnnn in the output table library.

Control Table Syntax

The general syntax is keywords followed by their associated values enclosed within parentheses: KEYWORD (value). If the series of keywords continues beyond one card, then a '-' must follow the last keyword on each card except the last. Operand values which are strings of text too long to fit on a single card can be continued by coding text through the last column of the card (CARDSIZE) and continuing in column one of the next card. No continuation symbol is used in this case.

WILDCARD usage

This facility supports positional wildcard characters in text messages.

Text Messages: You may place multiple strings of wildcard characters in WTO, WTOR and COMMAND text. When LOCO matches the wildcard characters in the actual console message against the LOCO table entries, LOCO ignores (accepts) any text in the wildcard positions.

Example: ABC****HIJ

LOCO will look for a match on ABC in the first three positions and a match on HIJ for the 8th through 10th positions. It will ignore (accept) any text in positions 4 through 7.

Variable symbol usage

There is also a set of variable symbols which may be keyed in the ACTION operand value. These variable symbols will be replaced with their real time values at the time of execution of the ACTION. The variable symbols are:

$Variable	Format	Max Length	Type
$TIME	hh:mm:ss	8	global
$DATE	mm/dd/yy	8	global
$DATE2	yyddd	5	global
$DATE3	yymmdd	6	global
$DATE4	mm/dd/yyyy	10	global
$DATE5	yyyyddd	7	global
$DATE6	yyyymmdd	10	global
$WEEKDAY	xxxxxxxxx	9	global
$CPUID	cccc	4	global

$UV#1	xxx... ...	120	global
...	xxx... ...	120	global
...	xxx... ...	120	global
$UV#32	xxx... ...	120	global
$R#1	rr	4	global
...	rr	4	global
...	rr	4	global
$R#16	rr	4	global
$TEXT	xxxx......	120	local
$MINOR	xxxx......	80	local
$JOBNAME	jjjjjjjj	8	local
$STEPNAME	ssssssss	8	local
$JOBID	JOB#####	8	local
$TSOID	ttttttt	7	local
$ORIGIN	xxxxxxxx	8	local
$EXITEXT	xxxx......	120	local

These $var symbols may be keyed in the ACTION operand value, and they will be replaced with their real time values at the time of execution of the ACTION. These variable symbols will have their names replaced by their exact current length of data; thus the resulting text string will expand or contract accordingly.

Optionally any of the $vars may be immediately followed by a substring expression specifying the beginning position and length of a portion of the $var value to be extracted for substitution. The format is $var(beg,len).

```
Examples: If $JOBNAME has the value 'ABCDEFGH' then
                  $JOBNAME(1,8) is 'ABCDEFGH'
                  $JOBNAME(1,3) is 'ABC'
                  $JOBNAME(7,1) is 'G'
```

The $vars have their values set as follows:

$CPUID: This value is set to the 4 digit value of the MVS CPU id field at LOCOMAIN startup time.

$TIME: This value and the various $DATEs are set to current real time values as LOCOMAIN is dispatched due to a WTO or @(time) triggering event.

$WEEKDAY: This value is set to the name of the current weekday

$UV#1 - #UV#32: These 32 generalized user variables and their respective values are established by the SET operand in STARTUP, WTO, WTOR, COMMAND or @(time) statements.

$R#1 - $R#16: These 16 specific user variables automatically retain and update WTOR reply numbers for deferred replys. Their values are established by the SET operand in the WTOR or PRIME statements.

$TEXT: This character string is set to the WTO, WTOR or COMMAND text which triggered the ACTION. By using several $TEXT substrings in the ACTION value, it is possible to substitute any combination of originating text into the ACTION text. Notice that whenever a series of ACTIONSs is interrupted by a WAITEXT, the value of $TEXT is reset to that of the most current WTO which triggers the WAITEXT condition.

$MINOR: This character string is set to the MINOR line text of the WTOwhich triggered the ACTION.

$JOBNAME: This value is set to the JCL jobname of the job, task or TSO user which issued the triggering activity.

$STEPNAME: This value is set to the JCL stepname of the job, task or TSO user which issued the triggering activity.

$JOBID: This value is set to the JES jobid of the job, task or TSO user which issued the triggering activity. The $JOBID will begin with JOB, STC or TSU followed by a 5 digit number.

$TSOID: This value is set to the TSO user id who submitted the job which issued the activity which triggered the ACTION.

$ORIGIN: This value is set to the JES sysid (field CTXTSYSN) of the system which issued the WTO(R) which triggered the ACTION.

$EXITEXT: This character string is set by User Exit programs. See SPECIAL TECHNIQUES section on EXIT PROGRAMS for details.

The following table depicts which $variables can participate in ACTIONS for specific LOCO control Table statements.

$Variable	TOD	WTO/	WTOR	PRIME	COMMAND	EVENT	STARTUP/
		MAJOR					SHUTDOWN
$TIME	X	X	X	X	X	X	X
$DATE	X	X	X	X	X	X	X
$DATE2	X	X	X	X	X	X	X
$DATE3	X	X	X	X	X	X	X
$DATE4	X	X	X	X	X	X	X
$DATE5	X	X	X	X	X	X	X
$DATE6	X	X	X	X	X	X	X

$WEEKDAY	X	X	X	X	X	X	X
$CPUID	X	X	X	X	X	X	X
$UV#nn	X	X	X	X	X	X	X
$R#nn	X	X	X	X	X	X	X
$EXITEXT	X	X	X	X	X	X	X
$TEXT		X	X		X		
$MINOR		X					
$JOBNAME		X	X	X		X	
$STEPNAME						X	
$JOBID		X	X	X		X	
$TSOID		X	X	X		X	
$ORIGIN		X	X	X			

DEFINE Statement

This statement provides LOCO with global information about this particular table. Normally the DEFINE statement is the first card of the LOCO table definition.

```
DEFINE        PREFIX(value)                    optional
              WILDCARD(value)                  optional
              SLEEP(value)                     optional
              CARDSIZE(number)                 optional
              ROUTE(values)                    optional
              QLIMIT(number)                   optional
```

PREFIX: The value is a single character to be used to identify LOCO commands and distinguish them from MVS or JES commands. This same character must be the first character of each defined LOCO command in COMMAND statements. The default is @.

NOTE: Please note that @ will be inappropriate for LOCO command prefix if your installation uses NCCF. In that case pick a different special character for the LOCO command prefix, say %.

WILDCARD: The value is a single character used in text values to represent the occurrence of any character for matching purposes. The default is *.

SLEEP: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. This keyword provides a global default for the SLEEP keyword of the WTO statement. If SLEEP is not explicitly coded on the DEFINE statement, then the global SLEEP default is zero 0.0) seconds.

CARDSIZE: The number is limited to a maximum value of 80. It determines the number of characters in an input statement that will be scanned by LOCOBLDR. A typical value would be 72 so that the statement cards may have sequence numbers. The default is 72.

ROUTE: The values are a series of routing and descriptor codes expressed as Rn and Dn respectively. The numeric portions range from 1 through 16 for descriptors and 1 thru 128 for routes and correspond to the values described in the IBM manual Routing and Descriptor Codes. Any combination of codes may be keyed separated by commas. The codes specified in the ROUTE operand

```
          here will override the default of (R2,D7) for LOCO
          messages produced by ACTION (@+.....).
QLIMIT:   The number is the upper limit on backlogged WTO/WTORs
          which LOCO has stored in either CSA (DS=N) or the
          LOCOMAIN dataspace (DS=Y) waiting on LOCOMAIN
          processing.  When the count of messages waiting for
          processing by LOCOMAIN reaches this number, succeeding
          WTOs/WTORs are lost to LOCO.   The default value is
          1024.  A larger number, may be necessary to handle
          large bursts of WTOs when LOCOMAIN is slow to dispatch,
          e.g. during VTAM initialization or when LOCOMAIN does
          not reside in a special, high priority performance
          group.  If the LOCO WTO/WTORs are held in CSA (DS=N),
          the QLIMIT make be adjusted downward so that LOCO can
          only utilize a maximum of 50% of the CSA available at
          table load time.
```

STARTUP Statement

This statement delineates an action or series of actions to be performed by LOCO as it originally starts execution. STARTUP is ideal for IPL activities. If a table is loaded into LOCO during execution (by the @LOAD command), the STARTUP actions are NOT performed.

```
STARTUP       ACTION(text)                      required
              CPUID(cccc)                       optional
              LOTIME(time)                      optional
              HITIME(time)                      optional
              DATE(date)                        optional
              LODATE(date)                      optional
              HIDATE(date)                      optional
              WEEKDAY(mtwtfss)                  optional
              WAITIME(time)                     optional
              WAITEXT(text)                     optional
              RESET(time)                       optional
              SET(#UV#nn=text)                  optional
```

ACTION: The text is one of two types of commands:

1. an MVS command,
2. a LOCO built-in command.

The ACTION keyword may be repeated on the STARTUP statement as many times as needed. At least one ACTION must be coded. The ACTION text may contain global $vars for real time substitution.

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. STARTUP actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The Filter condition may be reversed by preceding the cpu id value by the - (minus) symbol. E.G. CPUID(-CPUA).

LOTIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. STARTUP actions will only be performed on or after that particular time of day.

HITIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. STARTUP actions will only be performed on or before that particular time of day.

Note: HITIME and LOTIME may be specified together to define a time range.

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. STARTUP actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. STARTUP actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. STARTUP actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through 50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; STARTUP actions will NOT be performed on weekday positions which contain the character '.' Example WEEKDAY(M.W.F..) would cause the STARTUP actions to be performed on Monday, Wednesday and Friday only.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. If both are coded, the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are *functionally connected by a logical OR operator.*

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT without executing any of the following ACTIONs.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global $vars and blanks) will become the current value of the target user $var preceding the = symbol.

SHUTDOWN Statement

This statement delineates an action or series of actions to be performed by LOCO as it terminates execution.

```
SHUTDOWN      ACTION(text)                      required
              CPUID(cccc)                       optional
              LOTIME(time)                      optional
              HITIME(time)                      optional
              DATE(date)                        optional
              LODATE(date)                      optional
              HIDATE(date)                      optional
              WEEKDAY(mtwtfss)                  optional
              WAITIME(time)                     optional
              WAITEXT(text)                     optional
              RESET(time)                       optional
              SET(#UV#nn=text)                  optional
```

ACTION: The text is one of two types of commands:

1. an MVS command
2. a LOCO built-in command

The ACTION keyword may be repeated on the SHUTDOWN statement as many times as needed. At least one ACTION must be coded. The ACTION text may contain global $vars for real time substitution.

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. SHUTDOWN actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The Filter condition may be reversed by preceding the cpu id value by the - (minus) symbol. E.G. CPUID(-CPUA).

LOTIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. SHUTDOWN actions will only be performed on or after that particular time of day.

HITIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. SHUTDOWN actions will only be performed on or before that particular time of day.

Note: HITIME and LOTIME may be specified together to define a time range.

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. SHUTDOWN actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. SHUTDOWN actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. SHUTDOWN actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through 50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; SHUTDOWN actions will NOT be performed on weekday positions which contain the character '.' Example WEEKDAY(M.W.F..) would cause the SHUTDOWN actions to be performed on Monday, Wednesday and Friday only.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. If both are coded, the second ACTION will be performed as soon as EITHER wait *condition is satisfied. The wait keywords are functionally connected by a logical OR operator.*

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT without executing any of the following ACTIONs.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global $vars and blanks) will become the current value of the target user $var preceding the = symbol.

WTO Statement

The text is any string of characters (including the WILDCARD). This statement delineates an action or series of actions to be performed by LOCO whenever a WTO is issued through the MVS system matching the coded text.

```
 WTO          (text)
              ACTION(text)                     optional
              SUPPRESS                         optional
              DELETE                           optional
              REROUTE(routes)                  optional
              LOTIME(time)                     optional
              HITIME(time)                     optional
              DATE(date)                       optional
              LODATE(date)                     optional
              HIDATE(date)                     optional
              WEEKDAY(mtwtfss)                 optional
              JOBNAME(value)                   optional
              JOBTYPE(value)                   optional
              CPUID(cccc)                      optional
              TSOID(ttttttt)                   optional
              ORIGIN(value)                    optional
              THRESHOLD(count)                 optional
              SLEEP(time)                      optional
              WAITIME(time)                    optional
              WAITEXT(text)                    optional
              RESET(time)                      optional
              SET($UV#nn=text)                 optional
```

ACTION: The text is one of the two types of commands:

1. an MVS command,
2. a LOCO built-in command

The ACTION keyword may be repeated on the WTO statement as many times as needed. At least one ACTION must be coded. The ACTION text may contain global or local $vars for real time substitution.

SUPPRESS: This operand will cause the WTO to be routed to the hardcopy log only. It will not appear on any MVS console.

DELETE: This operand will cause the WTO to completely disappear.

REROUTE: This operand will change the routing and descriptor codes of the original WTO. The routing codes are keyed as Rn where n ranges from 1 through 128; descriptor codes are keyed similarly as Dn where n ranges from 1 through 16. Multiple routes and descriptors may be keyed in any order separated by commas. The actions implied by these codes are described in IBM's manual titled Routing and Descriptor Codes. Example: REROUTE(D3,R7)

LOTIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. WTO actions will only be performed on or after that particular time of day.

HITIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. WTO actions will only be performed on or before that particular time of day.

Note: HITIME and LOTIME may be specified together to define a time range.

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTO actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTO actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTO actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through 50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; The WTO actions will NOT be performed on weekday positions which contain a period. Example WEEKDAY(.....SS) would cause the WTO actions to be performed on Saturday and Sunday only.

JOBNAME: The value must follow the MVS JCL requirements for jobnames. It is a filter keyword. WTO actions will be performed only for MVS jobs with matching jobnames which issue the WTO text. The JOBNAME value may contain WILDCARD characters. The ACTION text may contain global or local $vars for real time substitution. The filter condition may be reversed by preceding the jobname value by the - (minus) symbol. E.G. JOBNAME(-FREDSJOB).

JOBTYPE: The jobtype must be one of the following strings: STC, JOB, TSU, MSTR, JES2, JES3. It is a filter keyword. WTO actions will only be performed for WTOS issued by subsystem (MSTR, JES2, JES3) or type of job (STC,JOB, TSU). The filter condition may be reversed by preceding the jobtype value by the - (minus) symbol. E.G. JOBTYPE(-STC).

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. WTO actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The filter condition may be reversed by preceding the cpu id value by the - (minus) symbol. E.G. CPUID(-CPUA).

TSOID: The tsoid is a 7 digit value representing the TSO user id of the submitter of the job which produced the WTO triggering event. It is a filter keyword. WTO actions will only be performed for WTOs from jobs submitted by that TSO user. The TSOID value may contain wildcard characters. The filter condition may be reversed by preceding the TSO id value by the - symbol. E.G. TSOID(-PC1).

ORIGIN: The origin is an 8 byte value representing the JES sysid from CTXTSYSN field of the CTXT. WTO actions will only be performed for WTOs from matching origins. This filter is particularly useful for JES3 environments.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

THRESHOLD:The count represents the number of times the WTO must be encountered before the ACTION is performed. THRESHOLD is reset each time the ACTION is performed.

SLEEP: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. This keyword will cause this WTO entry to be dormant for the specified amount of time after it has completed the entire series of ACTIONS. It is effectively removed from the table for the time period. The default is taken from the

DEFINE SLEEP statement. If SLEEP is not explicitly DEFINEd, it defaults to zero.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. In that case the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are functionally connected by a logical OR operator.

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT and recycle the LOCO table entry without executing any of the following ACTIONs. The table entry will then wait for the next occurrence of the original WTO text.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global or local $vars and blanks) will become the current value of the target user $var preceding the = symbol.

MAJOR Command

The text is any string of characters (including the WILDCARD). This statement delineates an action or series of actions to be performed by LOCO whenever a MULTI-LINE WTO is issued through the MVS system matching the coded text for both the major text line and the minor text line.

```
MAJOR        (text)
             MINOR(text)                      required
             MINOR#(value)                    required
             ACTION(text)                     optional
             SUPPRESS                         optional
             DELETE                           optional
             REROUTE(routes)                  optional
             LOTIME(time)                     optional
             HITIME(time)                     optional
             DATE(date)                       optional
             LODATE(date)                     optional
             HIDATE(date)                     optional
             WEEKDAY(mtwtfss)                 optional
             JOBNAME(value)                   optional
             JOBTYPE(value)                   optional
             CPUID(cccc)                      optional
             TSOID(ttttttt)                   optional
             ORIGIN(value)                    optional
             THRESHOLD(count)                 optional
             SLEEP(time)                      optional
             WAITIME(time)                    optional
             WAITEXT(text)                    optional
             RESET(time)                      optional
             SET($UV#nn=text)                 optional
```

MINOR: The text is matched to the minor data line of the MULTI-LINE WTO. The text data from the multi-line WTO is presented to LOCOMAIN as text pairs with each minor data line paired with the original major line. The ACTIONs are invoked only when both MAJOR and MINOR text matches the MLWTO. The $VARIABLE $TEXT will then contain the major text and $MINOR will contain the minor text for substitution into the ACTIONs. MINOR may contain wildcard characters. Either MINOR or MINOR# (but not both) must be specified.

MINOR#: The value is the sequential line number of the MLWTO minor line first minor text line is 1). In many cases minor lines of MLWTOs have no constant data to match for triggering actions. By specifying MINOR# the ACTIONs are triggered when the MAJOR text matches and the MINOR# line is present. The $VARIABLE $TEXT will then contain the major text and $MINOR will contain the minor text from the MINOR# data line for substitution into the ACTIONs. Either MINOR or MINOR# (but not both) must be specified.

ACTION: The text is one of the two types of commands: (1) an MVS command, (2) a LOCO built-in command. The ACTION keyword may be repeated on the WTO statement. At least one ACTION must be coded. The ACTION text may contain global or local $vars for real time substitution.

SUPPRESS: This operand will cause the WTO to be routed to the hardcopy log only. It will not appear on any MVS console. DELETE: This operand will cause the WTO to completely disappear.

REROUTE: This operand will change the routing and descriptor codes of the original WTO. The routing codes are keyed as Rn where n ranges from 1 through 128; descriptor codes are keyed similarly as Dn where n ranges from 1 through 16. Multiple routes and descriptors may be keyed in any order separated by commas. The actions implied by these codes are described in IBM's manual titled Routing and Descriptor Codes. Example: REROUTE(D3,R7)

LOTIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. WTO actions will only be performed on or after that particular time of day.

HITIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. WTO actions will only be performed on or before that particular time of day.

Note: HITIME and LOTIME may be specified together to define a time range.

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTO actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTO actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTO actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through 50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; The WTO actions will NOT be performed on weekday positions which contain a period. Example WEEKDAY(.....SS) would cause the WTO actions to be performed on Saturday and Sunday only.

JOBNAME: The value must follow the MVS JCL requirements for jobnames. It is a filter keyword. WTO actions will be performed only for MVS jobs with matching jobnames which issue the WTO text. The JOBNAME value may contain WILDCARD characters. The ACTION text may contain global or local $vars for real time substitution. The filter condition may be reversed by preceding the jobname value by the -(minus) symbol. E.G. JOBNAME(-FREDSJOB).

JOBTYPE: The jobtype must be one of the following strings: STC, JOB, TSU, MSTR, JES2, JES3. It is a filter keyword. WTO actions will only be performed for WTOS issued by subsystem (MSTR, JES2, JES3) or type of job (STC, JOB, TSU). The filter condition may be reversed by preceding the jobtype value by the - (minus) symbol. E.G. JOBTYPE(-STC).

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. WTO actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The filter condition may be reversed by preceding the cpu id value by the - (minus) symbol. E.G. CPUID(-CPUA).

TSOID: The tsoid is a 7 digit value representing the TSO user id of the submitter of the job which produced the WTO triggering event. It is a filter keyword. WTO actions will only be performed for WTOs from jobs submitted by that TSO user. The TSOID value may contain wildcard characters. The filter condition may be reversed by preceding the TSO id value by the - symbol. E.G. TSOID(-PC1).

ORIGIN: The origin is an 8 byte value representing the JES sysid from CTXTSYSN field of the CTXT. WTO actions will only be performed for WTOs from matching origins. This filter is particularly useful for JES3 environments.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

THRESHOLD: The count represents the number of times the WTO must be encountered before the ACTION is performed. THRESHOLD is reset each time the ACTION is performed.

SLEEP: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. This keyword will cause this WTO entry to be dormant for the specified amount of time after it has completed the entire series of ACTIONS. It is effectively removed from the table for the time period. The default is taken from the DEFINE SLEEP statement. If SLEEP is not explicitly DEFINEd, it defaults to zero.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. In that case the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are functionally connected by a logical OR operator.

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT and recycle the LOCO table entry without executing any of the following ACTIONs. The table entry will then wait for the next occurrence of the original WTO text.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1,

$UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global or local $vars and blanks) will become the current value of the target user $var preceding the = symbol.

WTOR Statement

The text is any string of characters (including the WILDCARD). This statement delineates a reply and possibly an action or series of actions to be performed by LOCO whenever a WTOR is issued through the MVS system matching the coded text. LOCO produces a confirmation message whenever it responds to a WTOR.

```
 WTOR          (text)
               REPLY(text)                      required
               REROUTE(routes)                  optional
               ACTION(text)                     optional
               LOTIME(time)                     optional
               HITIME(time)                     optional
               DATE(date)                       optional
               LODATE(date)                     optional
               HIDATE(date)                     optional
               WEEKDAY(mtwtfss)                 optional
               JOBNAME(value)                   optional
               JOBTYPE(value)                   optional
               CPUID(cccc)                      optional
               TSOID(ttttttt)                   optional
               ORIGIN(value)                    optional
               WAITIME(time)                    optional
               WAITEXT(text)                    optional
               RESET(time)                      optional
               SET($UV#nn=text)                 optional
               SET($R#)                         optional
```

REPLY: The text is any string of characters and may contain global or local $vars for real time substitution as the REPLY is issued. If the REPLY text is blank or has leading blanks, it must be enclosed in single quotes ' '. Example: REPLY (' ').

REROUTE: This operand will change the routing and descriptor codes of the original WTOR. The routing codes are keyed as Rn where n ranges from 1 through 128; descriptor codes are keyed similarly as Dn where n ranges from 1 through 16. Multiple routes and descriptors may be keyed in any order separated by commas. The actions implied by these codes are described in IBM's manual titled Routing and Descriptor Codes. Example: REROUTE(D2,R16).

ACTION: The text is one of two types of commands: (1) an MVS command, (2) a LOCO built-in command. The ACTION keyword may be repeated on the WTOR statement. The

ACTION text may contain global or local $vars for real time substitution

LOTIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. WTOR actions will only be performed on or after that particular time of day.

HITIME: The time is in the format of HH:MM:SS and represents a clock value. It is a filter keyword. WTOR actions will only be performed on or before that particular time of day.

Note: HITIME and LOTIME may be specified together to define a time range.

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTOR actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTOR actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. WTOR actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through 50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; The WTOR actions will NOT be performed on weekday positions which contain the character '.' Example WEEKDAY(MTW....) would cause the WTOR actions to be performed on Monday, Tuesday and Wednesday only.

JOBNAME: The value must follow the MVS JCL requirements for jobnames. It is a filter keyword; the WTOR actions will be performed only for MVS jobs with matching jobnames which issue the WTOR text. JOBNAME value may contain WILDCARD characters. The filter condition may be reversed by preceding the jobname value by the - (minus) symbol. E.G. JOBNAME(-OPER1**A).

JOBTYPE: The jobtype must be one of the following strings: STC, JOB, TSU, MSTR, JES2, JES3. It is a filter keyword, WTOR actions will only be performed for WTORS issued by subsystem (MSTR, JES2, JES3) or type of job (STC, JOB, TSU). The filter condition may be reversed by preceding the jobtype value by the - (minus) symbol. E.G. JOBTYPE(-JOB).

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. WTOR actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The filter condition may be reversed by preceding the cpu id value by the - symbol. E.G. CPUID(-CPUA).

TSOID: The tsoid is a 7 digit value representing the TSO user id of the submitter of the job which produced the WTO triggering event. It is a filter keyword. WTOR actions will only be performed for WTORs from jobs submitted by that TSO user. The TSOID value may contain wildcard characters. The filter condition may be reversed by preceding the TSO id value by the - symbol. E.G. TSOID(-PC1).

ORIGIN: The origin is an 8 byte value representing the JES sysid from CTXTSYSN field of the CTXT. WTOR actions will only be performed for WTORs from matching origins. This filter is particularly useful for JES3 environments.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. In that case the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are functionally connected by a logical OR operator.

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT and recycle the LOCO table entry without executing any of the following ACTIONs. The table entry will then wait for the next occurrence of the original WTOR text.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global or local $vars and blanks) will become the current value of the target user $var preceding the = symbol. SET is also used to automatically save and update the WTOR reply number in a reply $VAR. There are 16 reply $vars labeled $R#1, ..., $R#16. In this case only the reply $var label is specified in the SET operand. E.G. SET(R#2).

@ Command

The time is a clock value in the format HH:MM:SS. The actions will begin at the time of day specified.

```
@            (time)
              ACTION(text)                      required
              CPUID(cccc)                       optional
              DATE(date)                        optional
              LODATE(date)                      optional
              HIDATE(date)                      optional
              WEEKDAY(mtwtfss)                  optional
              REPEAT(time)                      optional
              UNTIL(time)                       optional
              WAITIME(time)                     optional
              WAITEXT(text)                     optional
              RESET(time)                       optional
              SET($UV#nn=text)                  optional
```

ACTION: The text is one of two types of commands: (1) an MVS command, (2) a LOCO built-in command. The ACTION keyword may be repeated on the @ statement. At least one ACTION must be coded. The ACTION may contain global $vars for real time substitution.

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. @time actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The filter condition may be reversed by preceding the cpu id value by the – symbol. E.G. CPUID(-CPUA).

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. @ actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. @ actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. @ actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through

50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; The @ actions will NOT be performed on weekday positions which contain a period. Example WEEKDAY(M...F..) would cause the @ actions to be performed on Monday and Friday only.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

REPEAT: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. This keyword will cause this @ entry to be repeated at the specified time interval until the time crosses the UNTIL time value, if UNTIL is specified. If UNTIL is not specified, as the time passes over the original @ time of the next day, it is synchronized back to the original time. Default is 24:00:00.

UNTIL: The time is of the format HH:MM:SS and represents a clock value. The UNTIL value marks the REPEAT limit. When the @ statement reaches its UNTIL limit, it is set to continue at the original time the following day.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. In that case the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are functionally connected by a logical OR operator.

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT and recycle the LOCO table entry without executing any of the following ACTIONs. The table entry will then wait for the time of the next REPEAT interval.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global $vars and blanks) will become the current value of the target user $var preceding the = symbol.

COMMAND Statement

The text is any string of characters (including the WILDCARD). This statement delineates an action or series of actions to be performed by LOCO whenever the COMMAND text is keyed on the MVS console. The first character of the text must be the LOCO command prefix as specified on the DEFINE statement.

```
COMMAND       (text)
              ACTION(text)                     required
              WAITIME(time)                    optional
              WAITEXT(text)                    optional
              RESET(time)                      optional
              SET(UV#nn=text)                  optional
```

ACTION: The text is either an MVS command or a LOCO built-in command. The ACTION keyword may be repeated on the COMMAND statement. At least one ACTION must be coded. The ACTION text may contain global $vars for real time substitution.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. In that case the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are functionally connected by a logical OR operator.

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT and recycle the LOCO table entry without executing any of the following ACTIONs. The table entry will then wait for the next occurrence of the original COMMAND text.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global or local $vars and blanks) will become the current value of the target user $var preceding the = symbol.

PRIME Command

The text is any string of characters (including the WILDCARD). This statement specifies that LOCO is to determine the reply number for the outstanding (already issued) WTOR text at the time of LOCO startup or at the time of a @LOAD command, and place the reply number in the the user's reply $VAR (as specified by the SET command). The PRIME statement is used in conjunction with the WTOR SET operand to insure that WTORs issued before LOCO startup are maintained consistently with WTORs issued after LOCO startup. See SPECIAL TECHNIQUES Section.

```
PRIME         (text)
              SET($R#nn)                       required
              JOBNAME(value)                   optional
```

SET: Set is an ACTION keyword used to save the WTOR reply number in a reply $var. There are 16 reply $vars labeled $R#1, ..., $R#16.

JOBNAME: The value must follow the MVS JCL requirements for jobnames. It is a filter keyword; the PRIME action will be performed only for MVS jobs with matching jobnames which issue the WTOR text. JOBNAME value may contain WILDCARD characters.

EVENT(endofstep) Statement

The endofstep event is triggered when an MVS task terminates which has a matching jobname and stepname.

```
EVENT(ENDOFSTEP)
              JOBNAME(value)                  required
              STEPNAME(value)                 required
              ACTION(text)                    required
              JOBTYPE(value)                  optional
              CPUID(cccc)                     optional
              TSOID(ttttttt)                  optional
              DATE(date)                      optional
              LODATE(date)                    optional
              HIDATE(date)                    optional
              WEEKDAY(mtwtfss)                optional
              WAITIME(time)                   optional
              WAITEXT(text)                   optional
              RESET(time)                     optional
              SET($UV#nn=text)                optional
```

ACTION: The text is one of two types of commands: (1) an MVS command, (2) a LOCO built-in command. The ACTION keyword may be repeated on the @ statement. At least one ACTION must be coded. The ACTION may contain global $vars for real time substitution.

JOBNAME: The value must follow the MVS JCL requirements for jobnames. It is a filter keyword; the ENDOFSTEP actions will be performed only for terminating MVS tasks with matching jobnames. JOBNAME value may contain WILDCARD characters.

STEPNAME: The value must follow the MVS JCL requirements for stepnames. It is a filter keyword; the ENDOFSTEP actions will be performed only for terminating MVS tasks with matching stepnames. STEPNAME value may contain WILDCARD characters.

JOBTYPE: The jobtype must be one of the following strings: STC, JOB, TSU. It is a filter keyword, ENDOFSTEP actions will only be performed for terminating jobs of matching type.

CPUID: The cpuid is a four digit value representing the internal MVS id of the cpu. It is a filter keyword. ENDOFSTEP actions will only be performed when LOCO is invoked on that particular cpu. The CPUID value may contain wildcard characters. The filter condition may be reversed by preceding the cpu id value by the - symbol. E.G CPUID(-CPUA).

TSOID: The tsoid is a 7 digit value representing the TSO user id of the submitter of the job which terminated. It is a filter keyword. ENDOFSTEP actions will only be performed for terminating jobs submitted by that TSO user. The TSOID value may contain wildcard characters. The filter condition may be reversed by preceding the TSO id value by the - symbol. E.G. TSOID(-PC1).

DATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. ENDOFSTEP actions will only be performed on that particular date.

LODATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. ENDOFSTEP actions will only be performed on or after that particular date.

HIDATE: The date must follow the exact format of MM/DD/YY or MM/DD/YYYY. It is a filter keyword. ENDOFSTEP actions will only be performed on or before that particular date.

Note: HIDATE and LODATE may be specified together, but DATE may not be specified with either of them. If any of the above date operands is specified as MM/DD/YY, LOCO will provide the complete YYYY internally. If YY is in the range of 00 through 50, then the century will be set to 20; if YY is higher than 50, then the century will be set to 19.

WEEKDAY: The string of characters represents the seven days of the week beginning with Monday. It is a filter keyword; The ENDOFSTEP actions will NOT be performed on weekday positions which contain a period. Example WEEKDAY(M...F..) would cause the ENDOFSTEP actions to be performed on Monday and Friday only.

Note: Whenever multiple filter keywords are coded, all criteria must be satisfied in order for the actions to be performed. The filter keywords are functionally connected by a logical AND operator.

WAITIME: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A WAITIME may be coded after an ACTION causing a wait for that amount of time before proceeding to the next ACTION.

WAITEXT: The text is any string of characters (including the WILDCARD). A WAITEXT may be coded after an ACTION causing a wait until that text string is issued as a WTO through the MVS system. After the text is seen as a WTO, the next ACTION is performed.

Note: Both WAITIME and WAITEXT may be coded between ACTIONS. In that case the second ACTION will be performed as soon as EITHER wait condition is satisfied. The wait keywords are functionally connected by a logical OR operator.

RESET: The time is of the format HH:MM:SS and represents an amount of time, not a clock value. A RESET may be coded only after a WAITEXT and specifies an amount of time after which LOCO is to automatically terminate the WAITEXT and recycle the LOCO table entry without executing any of the following ACTIONs. The table entry will then wait for the time of the next ENDOFSTEP event.

SET: SET is an ACTION keyword used to insert values into user $VARS. There are 32 user $vars labeled $UV#1, $UV#2, ... $UV#32. The = symbol must immediately follow the user $VAR label. All text following the = symbol (including other global $vars and blanks) will become the current value of the target user $var preceding the = symbol.

CAFCCMD Statement

This statement delineates a CAFC action or series of CAFC actions to be performed by LOCO whenever the COMMAND text is keyed on the MVS console or encountered as an action of another LOCO command. The first character of the text must be the LOCO command prefix as specified on the DEFINE statement. The use of the CAFCCMD requires the presence of the NETEC CAFC product. CAFC is a separate product that can be licensed from NETEC International. LOCO utilizes the CAFC batch interface program to make CAFC request. The order of the keywords found on the CAFCCMD are important. Keywords found before the first ACTION keyword apply to the entire CAFCCMD. The RC, SESSION, and NORESPONSE keywords, found after an ACTION keyword apply only to that ACTION.
To fully understand the CAFC batch interface the user should review the CICS Application Control Facility Installation Manual.

CAFCCMD	(text)	required
	TIMEOUT(HH:MM:SS)	optional
	TRACE(Y/N)	optional
	LUTIME(HH:MM:SS)	optional
	LUMSG(nn)	optional
	DELAY(HH:MM:SS)	optional
	TERMDLAY(HH:MM:SS)	optional
	FREEDSN(Y/N)	optional
	DEFACB(Y/N)	optional
	DIRECT(Y/N)	optional
	EXCITIME(HH:MM:SS)	optional
	EXCIMSG(mm)	optional
	CONNTYPE(L/E)	optional
	PAUSE(HH:MM:SS)	optional
	RC(IGNORE/ACCUM)	optional
	CTLFILE(applid,cntl_file_dsname)	required
	ACTION(CAFC Command)	required
	RC(IGNORE/ACCUM)	optional
	SESSION(CONT/END)	optional
	NORESPONSE	optional

TIMEOUT: After LOCO issues a RECEIVE to CICS, it sets a timer with a default limit of 2 minutes. If CICS does not respond within 2 minutes, LOCO will terminate the request with the message 'WAIT TIME EXPIRED - CICS NO RESPONSE'. This situation is often related to poor CICS performance or to a data set being tied up by a user for an unusually long period of time. If your site frequently encounters this problem, you should increase the TIMEOUT value. The

TIMEOUT value is specified by coding the number of hours, minutes, and seconds to wait for a request to complete. TIMEOUT(00:01:00) request that CAFC wait 1 minute before considering the CICS as non-responsive. For a complete description of this parameter see the CICS Application File Control Facility Installation Manual. The default timeout value is 2 minutes.

TRACE: TRACE(Y) will cause the CAFCCMD to create a NETEC internal trace while executing the command. This parameter overrides the default trace setting which is determine by checking to see if trace is active in the LOCOMAIN task. If this parameter is omitted, tracing activity will default to the current trace status of the LOCOMAIN task.

LUTIME: Before attempting to establish a session with CICS, LOCO checks, via enqueues, the availability of the LU names specified in AFCT2016. If the end of the list is reached without finding an available (not busy) LU name, the program waits for an interval of time before reprocessing the list. The wait value specified must be in the following format: 00:00:00 - (hh:mm:ss) The default value is 2 minutes. You may specify an override wait interval at run time by coding the LUTIME(hh:mm:ss) keyword on the CAFCCMD statement For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

LUMSG: After LOCO reaches the end of the LU names list without finding an available LU name, LOCO checks the "message frequency" value. Each time the number of unsuccessful searches through the list of LUs equals the message "frequency value", LOCO writes a message. The "message frequency" value must be a 2-digit numeric value from 01-99. The default value is 03. You may specify an override message frequency at run time by coding the LUMSG(nn) keyword on the CAFCCMD statement. For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

DELAY: A time delay may be invoked by entering the DELAY keyword on the CAFCCMD statement. The DELAY is executed after the last CAFC request and the return code is zero. The format of the DELAY parameter is DELAY(hh:mm:ss). For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

TERMDLAY: A time delay for the Batch Terminal Processor (LU0) may be invoked to allow the started user transaction to complete before the VTAM session is terminated. The format of the TERMDLAY parameter is TERMDLAY(hh:mm:ss). For a complete description of this parameter see the

CICS Application File Control Facility Installation Manual.

FREEDSN: A CAFC Table File dsname switch may be invoked by the FREEDSN keyword parameter. If this parameter is set to "Y", the Table File data set name specified in the CAFCCMD CTLFILE keyword will be freed and the Table File data set name coded in the AFCT2016 table will be allocated and used by LOCO. The format of the FREEDSN is FREEDSN(Y). For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

DEFACB: If you wish to force the LOCO to use the default ACB name to connect to the target CICS regions, use the DEFACB keyword parameter. If this parameter is set to "Y", the LOCO will use 'CICSI62' to connect to all regions even though specific ACB names have been coded in the B/I Association Table, AFCT2016. The format of the DEFACB parameter is DEFACB(Y). For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

DIRECT: If you are certain a target CICS regions is down, you can specify the DIRECT parameter to CAFC through the DIRECT(Y) keyword on the CAFCCMD statement. CAFC will immediately apply the requests for this region to the CAFC Table File. The direct updates are performed without first checking that the CICS region is down. If the region is active and the Table file is already open, the requests to that region will fail with an open error. Subsequent requests to other regions will be processed if the respective CAFC Table files are available. DIRECT is useful if you need to synchronize or override the desired restart status for the next region startup. The format of the DIRECT parameter is DIRECT(Y). For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

EXCITIME: This parameter can be used to modify the time that LOCO will wait to acquire an EXCI receive session before attempting to acquire the session again. The time is specified as EXCITIME(hh:mm:ss).   The default value is 30 seconds. For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

EXCIMSG: This parameter can be used to modify the number of EXCITIME intervals that will be allowed to expire before a message will be issued indicating that LOCO has been waiting for a receive session. The default value is EXCIMSG=03 (issue a message every 3 intervals of 30 seconds i.e. 90 seconds). For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

CONNTYPE: This parameter can be used to modify the default connection type. Specify L for LU 6.2 or E for EXCI. The default value is CONNTYPE=L for LU 6.2. For a complete description of this parameter see the CICS Application File Control Facility Installation Manual.

RC: This parameter is used to specify if the return codes from the CAFC batch interface should be accumulated or ignored. The position of this operand in the CAFCCMD statement determines if it is the default or to be applied to a specific CAFC ACTION. If the RC operand is found in the CAFCCMD before any ACTION keywords, the RC is specifying the default for all CAFC ACTIONS found in the CAFCCMD. If the RC operand follows an ACTION, it applies only to that CAFC ACTION. Coding RC(ACCUM) is the equivalent of coding a “A” in column 72 of a CAFC control card and coding RC(IGNORE) is the equivalent of coding a “I” in column 72 of a CAFC control card. For a complete description of ignoring and accumulating return codes see the CICS Application File Control Facility Installation Manual.

SESSION: This parameter is used to specify if the LU0 transaction specified in the CAFCCMD ACTION filed is to be continued or is the final response. Coding SESSION(CONT) is the equivalent of coding a “C” in column 72 of a CAFC control card and coding SESSION(END) is the equivalent of coding a “E” in column 72 of a CAFC control card.

NORESPONSE: This parameter is used to specify if the LU0 transaction specified in the CAFCCMD ACTION field is not expected to give a response. Coding NORESPONSE is the equivalent of coding a “N” in column 71 of a CAFC control card. For a complete description of executing CICS transactions through the CAFC LU0 interface see the CICS Application File Control Facility Installation Manual.

CTLFILE: This field contains the CICS applid and the CAFC control file dataset name that the CAFC commands found in the ACTION key words are to be executed against. The applid is the 1 to 8 character VTAM application identifier of the target CICS system. The cntl_file_name is the fully qualified data set name of the CAFC control file of the target CICS system. The CTLFILE keyword can be repeated as many times as necessary.

ACTION: This field contains the CAFC command that is to be issued against the CICS system defined in the previous

CTLFILE field. The CAFC command formats can be found in the CAFC Users manual. The ACTION keyword can be repeated as many times as necessary.

Note: ACTIONS are applied to the CICS system described by the preceding CTLFILE keyword. By coding a sequence of CTLFILE keywords followed by ACTION keywords, multiple CAFC commands can be issued against multiple CICS regions as part of the same CAFCCMD defined command.

The following example will cause the DDNAME01 and DDNAME02 files to be opened in CICS99P and the DDNAME03 and DDNMAE04 files to be closed in CICS99X when the @CAFCTST commands is issue in LOCO.

```
CAFCCMD(@CAFCTST)
   CTLFILE(CICS99P,CICS99P.CAFC.CONTROL.FILE)
     ACTION(O,DD,DDNAME01)
     ACTION(O,DD,DDNAME02)
   CTLFILE(CICS99X,CICS99X.CAFC.CONTROL.FILE)
     ACTION(C,DD,DDNAME03)
     ACTION(C,DD,DDNAME04)
```

DBCTCMD Statement

This statement delineates a DBControl action or series of DBControl actions to be performed by LOCO whenever the COMMAND text is keyed on the MVS console or encountered as an action of another LOCO command. The first character of the text must be the LOCO command prefix as specified on the DEFINE statement. The use of the DBCTCMD requires the presence of the NETEC DB Control Batch interface program product. The DBControl Batch interface is a separate product that can be licensed from NETEC International.

```
DBCTCMD       (text)
              IMSID(ims_name)                    required
              ACTION(DBControl command)          required
              TRACE(Y/N)                         optional
              MAXRC(number)                      optional
              PSBNAME(name)                      optional
              RESPONSE(WAIT/NOWAIT)              optional
              WAITTIME(number)                   optional
              SUFFIX(xx)                         optional
```

IMSID: The IMSID field specifies the 1 to 4 character name of the DBControl system that LOCO is to connect to issue the DBControl commands found in the ACTION key words.

ACTION: The action field contains the DB Control command that is to be issued when the character string specified in the text field is entered as a LOCO command. The ACTION filed can be repeated as many times as needed.

TRACE: TRACE(Y) will cause the DBCTCMD to create a NETEC internal trace while executing the command. This parameter overrides the default trace setting which is determine by checking to see if trace is active in the LOCOMAIN task. If this parameter is omitted, tracing activity will default to the current trace status of the LOCOMAIN task.

MAXRC: This is the maximum return code that is allowable between actions commands. If the MAXRC value is exceeded further ACTIONS will be bypassed. The default value is 0.

PSBNAME: This field contains the PSBNAME to be used when issuing commands to the IMSID DB Control system. The default value is DBC9000.

RESPONSE: This field specifies as to whether or not to wait for a positive response to commands specified in LOCO

ACTIONS. WAIT indicates to wait for a response. NOWAIT specifies to continue without waiting for command completion. The default value is NOWAIT.

WAITTIME: This field contains the time to wait for a command to complete if RESPONSE(WAIT) is specified. The default value is 2 minutes.

SUFFIX: This field contains the suffix for the DFSPZP module to use when connecting to DB Control. The default value is 00.

Compressing LOCO Table and Submit Libraries

To prevent the possibility of compressing the table library while an @LOAD command or LOCOBLDR process is active, all use of the LOCO table library and LOCO submit library through the LOCO subsystem is protected via operating system serialization techniques. A compress of the table library performed by an external batch job, utilizing IEBCOPY, while LOCOMAIN is loading any table member could readily produce a damaged table load by with unknown but potentially disastrous results. If you need to compress the LOCO table library while LOCOMAIN is running, utilize the internal LOCO @COMPRESS command.

Likewise, a compress of the submit library performed by an external batch job utilizing IEBCOPY, while the LOCOMAIN is submitting a batch job or while the TSO/ISPF interface is saving a JCL member could produce a damaged submit library with unknown but potentially disastrous results. If you need to compress the LOCO submit library while LOCOMAIN is running, utilize the internal LOCO @COMPRESS command.

Pseudo-WTO Facility

LOCO has a facility to produce pseudo-WTOs which will interact with LOCOMAIN as if they were real WTOs. The pseudo-WTOs are created from parm or card-image input to program LOCOINTF.

By including program LOCOINTF as a conditional job step in your production job streams, it is easy to produce standardized messages instructing operations personnel to notify the responsible programmers. Then whenever application responsibility or programming personnel changes occur, a simple update to the LOCO control table will correct all notify messages.

Suppose that programmer I. Don Tell is responsible for all of the many payroll job streams. In each payroll job stream insert a conditional job step (perhaps in multiple places) similar to:

```
     //TELLDON    EXEC    PGM=LOCOINTF,COND=(O.GT.STEP3),
     //                   PARM='SSN=xxxx,MSG=''FAIL PAYROLL STEP 3'''
```

The xxxx is the subsystem name you have chosen for LOCO, and the corresponding LOCO control table entry would be:

```
WTO(FAIL PAYROLL ********) -
     ACTION(@+(R2,D2)PAYROLL JOB $JOBNAME FAILED IN ********) -
     ACTION(@+(R2,D2)NOTIFY PROGRAMMER DON TELL AT (123)456-7890)
```

Now suppose that in a moment of indiscretion Don Tell did tell and was fired because of it. In order to update the notify messages for the entire payroll system, only the second ACTION in the LOCO control table entry need be changed to reflect the name and phone number of Don Tell's replacement.

The Pseudo WTO facility has many applications for on-line environments. It is an efficient interface between batch jobs and TP control programs such as VTAM, NCCF, CICS, IMS, TSO etc. The message that the batch job step posts to LOCO can trigger: (1) on-line transactions, (2) CICS CEMT commands, (3) IMS MTO commands,(4) notifications to TSO users (5) a reply to an outstanding WTOR. Batch job communications with SCPs that use WTORs are extremely easy to implement because LOCO automatically captures and updates reply numbers for key outstanding WTORs. In addition LOCO will automatically log your master console onto each TP region that can accept MVS modify commands. With these standard techniques, your LOCO actions can utilize WTOR responses or modify commands to communicate with TP components.

In a production environment with or without an automated scheduling system, additional job steps are simply added to job streams and condi- tionally executed or bypassed. Applications include: closing and free- ing needed files, changing file status to read only, disabling transa- ctions during batch updates, closing/opening data bases and placing terminals on or off-line. The Pseudo WTO facility is invaluable in shut down procedures for on-line systems and networks.

Inserting LOCO Psuedo-WTOs in Batch Job Streams

The LOCOINTF program can be run as a conditional jobstep within your jobstreams to issue psuedo-WTOs to LOCOMAIN. If a single psuedo-WTO is to issued, it can be specified in the PARM; if multiple psuedo-WTOs are to be issued, they are provided as SYSIN data.

Example 1:

```
//TELLOCO1 EXEC   PGM=LOCOINTF,
//                PARM='SSN=xxxx,MSG=''PAYROLL STEP 3 FAILED''',
//                COND=(O.GT.STEP3)
```

Corresponding LOCO table statements might be:

```
WTO (PAYROLL STEP ** FAILED) -
    ACTION(@+(R2,D2)PAYROLL FAILED IN STEP **)
```

Example 2:

```
//TELLOCO2 EXEC  PGM=LOCOINTF,
//               PARM='SSN=xxxx',
//               COND=(O.GT.STEP3)
//SYSIN DD *
PAYROLL STEP  3 FAILED
NOTIFY:  BO DEREK
```

Corresponding LOCO table statements might be:

```
WTO (PAYROLL STEP ** FAILED) -
     ACTION(@+(R2,D2)PAYROLL FAILED IN STEP **)
WTO (NOTIFY:**************************) -
     ACTION(@+(R2,D2) CALL PROGRAMMER: ********************)
WTO (NOTIFY: BO DEREK) -
     ACTION(@+(R2,D2) PHONE NUMBER:  222-4444)
WTO (NOTIFY: BILLY KIDD) -
     ACTION(@+(R2,D2) PHONE NUMBER:  111-9999)
WTO (NOTIFY: FREDDY FIDDLER) -
     ACTION(@+(R2,D2) PHONE NUMBER:  333-7777)
```

Note: You must tell LOCOINTF which LOCO to pass the messages to by the PARM SSN=xxxx.

Creating Custom LOCO Actions via User Exit Programs

User exit programs must be written in Assembler. The calling sequence is standard:

```
     R1   ---> one word parm list
     R13  ---> SAVEAREA in LOCOMAIN
     R14  ---> return point in LOCOMAIN
     R15  ---> beginning of exit program
```

```
parm list ---> parm (244 byte area)
       L1      DS    H      length of input data (max 110 bytes)
       D1      DS    CL120  input data
       L2      DS    H      length of output data (max 120 bytes)
       D2      DS    CL120  output data
```

If the following ACTION is encountered in an LOCO control table:

```
     ACTION(@@(USERPROG)ABCDEF)
```

then USERPROG will be loaded into the LOCOMAIN address space and retained there. L1 will be set to H'6', D1 will be set to C'ABCDEF' with trailing blanks. L2 will be set to H'0', D2 will be set to blanks. R1 will be set to a one word parm list, and that word will contain the address of L1. The address of USERPROG will be loaded into R15 and a BALR R14,R15 will be issued.

When control is returned to LOCOMAIN, if D2 is none-zero, then the data in D2 is set into the Variable $EXITEXT and is available to subsequent ACTIONs. USERPROG must be linked as APF authorized and loaded from an APF authorized library so that LOCOMAIN will retain its authorization.

The following sample LOCO user exit programs are included in the LOCO distribution source library:

```
      LOCOEXIT  ALC exit program
```

Operations Guide

Operational Commands

LOCO has a variety of built-in commands to control its activities. A special character that is defined by the user on the DEFINE PREFIX statement prefixes each LOCO command. This document will use the default value of @. The LOCO commands are keyed on the MVS console.

The general syntax for LOCO commands is a command verb followed by none or more operands. The operands may be separated from the verb (and from each other) by a comma or by blanks.

COMMAND	PURPOSE
@HELP	display list of LOCO commands on console
@>	submit job stream through JES internal reader
@STOP	terminate LOCO execution
@STATUS	display information about LOCO on console
@LOAD	change to a new control table
@DEACTIVATE	temporarily stop LOCO functions
@ACTIVATE	continue LOCO functions
@HOLD	suspend LOCO time based functions
@RELEASE	resume LOCO time based functions
@RESET	recycle all table entries WAITing TEXT
@ZERO	set LOCO statistical counters to zero
@+	write a message to the console
@$	send a message to a TSO user
@%	print a message to SYSPRINT
@!	write a message to CRITICAL file
@@	call user written exit program
@REFRESH	load new copy of exit program

@HELP Command

The HELP command will list the LOCO commands on the MVS console. There are no operands.

COMMAND	OPERANDS
@HELP	none

@> Command

The SUBMIT command will select a JCL job stream from the LOCO job library and submit it through the JES internal reader. The operand is a member name located in the LOCO SUBMIT library. Example: @>MYJOB

COMMAND	OPERANDS
@>	jobname

@STOP Command

The STOP command will immediately terminate the LOCO task. There are no operands.

COMMAND	OPERANDS
@STOP	none

@STATUS Command

The STATUS command will display several messages on the MVS console containing information about the current LOCO status. The name and compile time-stamp of the current LOCO control table is displayed; the activation status and processing counts of WTO, WTOR and TIME events are displayed; the last LOCO activity is displayed. There are no operands.

COMMAND	OPERANDS
@STATUS	none

@LOAD Command

The LOAD command will immediately replace the current control table with the one specified as tabname. The operand TABLE= may alternately be specified as MEMBER=. If new tabname is not found, the current table remains in effect.

COMMAND	OPERANDS
@LOAD	TABLE=tabname

@DEACTIVATE Command

The DEACTIVATE command will cause LOCO to ignore the specified type of triggering activities until reACTIVATEd. Time initiated events WILL be discarded as they age. Any combination of the operands may be specified. If none are keyed then LOCO will deactivate all four triggering types. An LOCO status will show the deactivated triggering activities as 'PAUSED'.

COMMAND	OPERANDS
@DEACTIVATE	WTO
	WTOR
	TIME
	EVENT

@ACTIVATE Command

The ACTIVATE command will cause LOCO to resume processing the specified type of triggering activities. Any combination of the operands may be specified; if none are keyed then LOCO will activate all four triggering types. An LOCO status will show the activated events as 'ACTIVE'.

COMMAND	OPERANDS
@ACTIVATE	WTO
	WTOR
	TIME
	EVENT

@HOLD Command

The HOLD command will cause LOCO to cease processing time events. The time events will not be discarded but will queue up until released by @RELEASE. The TIME operand must be specified. An LOCO status will show the time event as 'HELD'.

COMMAND	OPERANDS
@HOLD	TIME

@RELEASE Command

The RELEASE command will cause LOCO to resume processing time events. All time events which were queued will be performed immediately. An LOCO status will show the time event as 'ACTIVE'.

COMMAND	OPERANDS
@RELEASE	TIME

@RESET Command

The RESET command will cause every LOCO table entry which is in a state of WAITEXT to terminate the WAITEXT and to be recycled to its original wait condition without executing any of the following ACTIONs. The RESET command may be keyed from the MVS console, but its most logical use would be automatic execution at some appropriate time of day from a time entry in the LOCO table.

```
          Example:  @(23:59) ACTION (@RESET)
```

COMMAND	OPERANDS
@RESET	none

@ZERO Command

The ZERO command will immediately set all LOCO statistical counters
to zero.

COMMAND	OPERANDS
@ZERO	none

@+ Command

The WTO command may not be keyed on the MVS console. It is for use as an ACTION in an LOCO control table. It may be used to produce additional explanatory messages for some particular WTO. It may be used in addition to a REPLY to produce an audit trail or confirmation of LOCO automatic replies.

COMMAND	OPERANDS
@+	message text
@+	(Rn,Dn) message text

The second format allows explicit routing and descriptor codes to be specified. The routing codes are keyed as Rn where n ranges from 1 through 128; descriptor codes are keyed similarly as Dn where n ranges from 1 through 16. The actions implied by these codes are described in IBM's manual titled Routing and Descriptor Codes.

The syntax is:

- multiple routes and descriptors may be keyed in any order separated by commas.
- the list of routing and descriptor codes must be enclosed in parentheses immediately following the @+.

```
Example:  ACTION(@+(R7,R3,D5)message)
          ACTION(@+(R2,D2)non-deletable message).
```

If no routing or descriptor codes are specified for the @+ command then (R2,D7) is used as an initial default. This default may be changed by the ROUTE operand on the DEFINE statement.

@$ Command

The SEND command may not be keyed on the MVS console. It is for use as an ACTION in an LOCO control table. It may be used to send a message to a specific TSO user or to the implied owner of an executing job.

COMMAND	OPERANDS
@$	message text
@$	(ttttttt) message text

The second format allows an explicit destination for the message text to be specified as TSO user id ttttttt. In the first format the dest- ination is implied (defaulted to) as the value of the variable $TSOID.

@% Command

The PRINT command may not be keyed on the MVS console. It is for use as an ACTION in an LOCO control table. It may be used to print a message to the SYSPRINT data set.

COMMAND	OPERANDS
@%	message text

@! Command

The CRITICAL command is for use as an ACTION in an LOCO control table. It may be used to write a message describing a critical situation or condition to the file defined by the //CRITICAL DD statement in the LOCOMAIN JCL. The CRITICAL file may be reviewed periodically thru TSO or some other software product which can browse a file online. The CRITICAL file remains closed between @! commands.

COMMAND	OPERANDS
@!	message text

If the two byte immediately following the ! character is a #1, #2 or #3, then the message is written to CRITCAL1, CRITCAL2 or CRITCAL3 ddname respectively. This facility allows different levels of intensity of critical messages to be segregated into different files.

@@ Command

The CALL command is for use as an ACTION in an LOCO control table. It may be used to invoke a user written exit program.

```
          Example:  ACTION (@@(yourprog)abcdefgh)
```

COMMAND	OPERANDS
@@(pppppppp)	text

The text operand is passed to the exit program, and data may be returned into variable symbol $EXITEXT which may be used in subsequent ACTIONs. See the SPECIAL TECHNIQUES chapter for details on EXIT PROGRAMS.

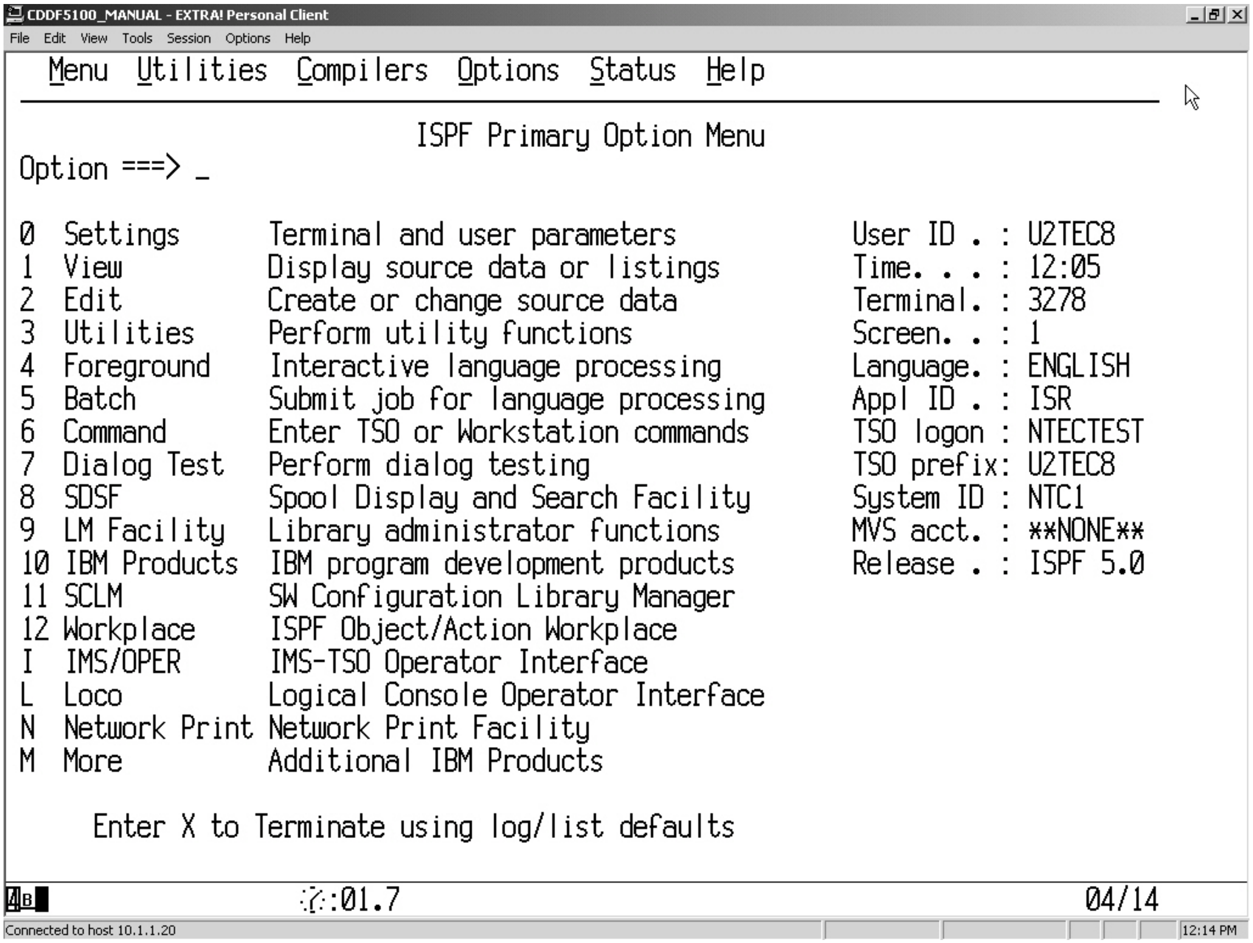
@REFRESH Command

The REFRESH command will cause the specified user exit program to be reloaded so that a newly assembled exit program can be implemented without terminating LOCO. If no operand is specified, then all user exit programs will be reloaded.

COMMAND	OPERANDS
@REFRESH	exit program name
@REFRESH	

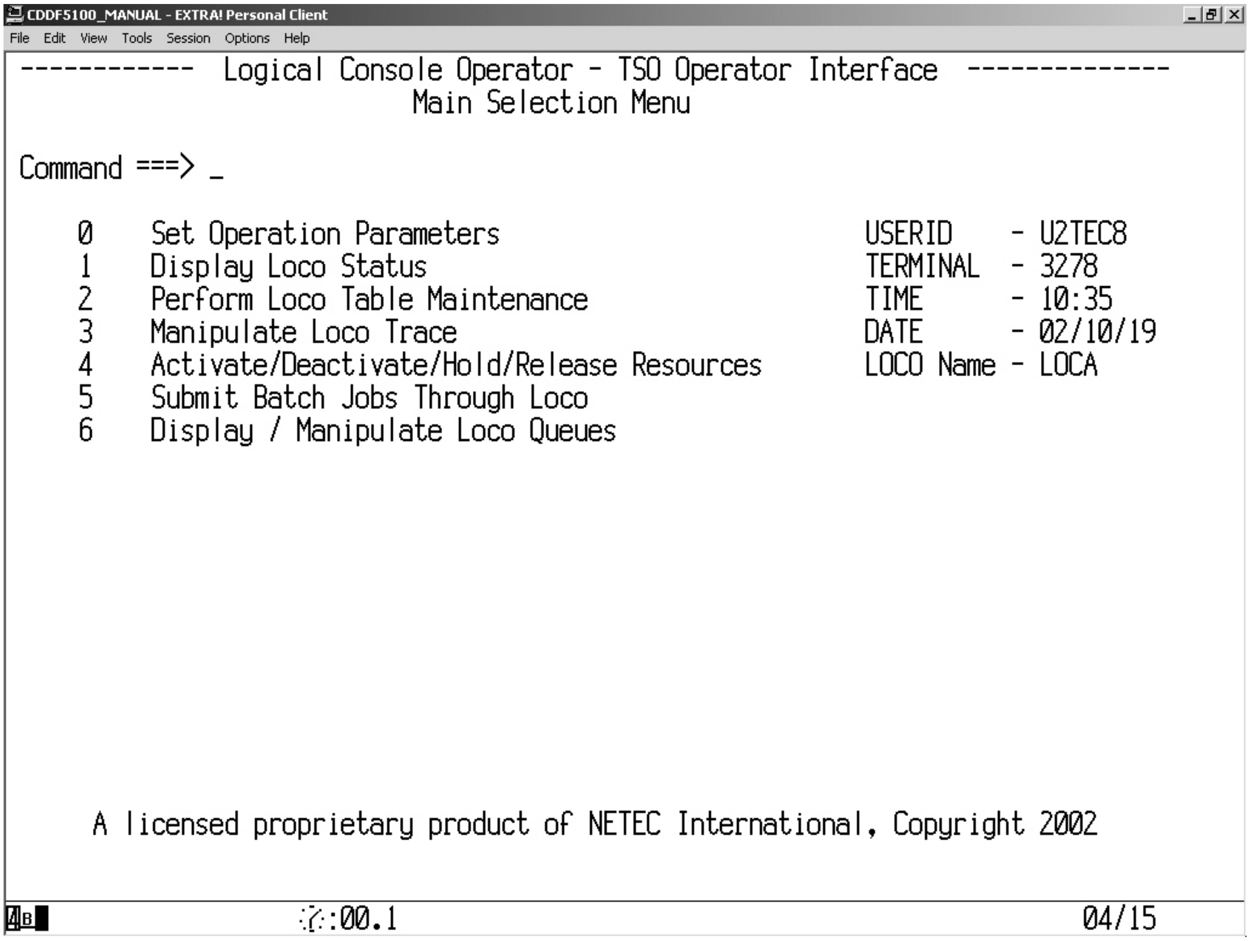
TSO Users Guide

A new feature in LOCO 3.3 is the ability to maintain and control LOC from a TSO interface. ANY TSO user that is LOGONed to the same MVS image where LOCO is executing and has sufficient security authorization may interactively maintain LOCO tables and control LOCO functions. The LOCO TSO interface is implemented as an ISPF application. The following screen should look something like your primary ISPF options screen. In this example the user would select option L to invoke the LOCO TSO interface.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 10.1.1.20
12:14 PM

```
   Menu  Utilities  Compilers  Options  Status  Help
 ------------------------------------------------------------------------------
                            ISPF Primary Option Menu
 Option ===> _

 0  Settings      Terminal and user parameters            User ID . : U2TEC8
 1  View          Display source data or listings         Time. . . : 12:05
 2  Edit          Create or change source data            Terminal. : 3278
 3  Utilities     Perform utility functions               Screen. . : 1
 4  Foreground    Interactive language processing         Language. : ENGLISH
 5  Batch         Submit job for language processing      Appl ID . : ISR
 6  Command       Enter TSO or Workstation commands       TSO logon : NTECTEST
 7  Dialog Test   Perform dialog testing                  TSO prefix: U2TEC8
 8  SDSF          Spool Display and Search Facility       System ID : NTC1
 9  LM Facility   Library administrator functions         MVS acct. : **NONE**
 10 IBM Products  IBM program development products        Release . : ISPF 5.0
 11 SCLM          SW Configuration Library Manager
 12 Workplace     ISPF Object/Action Workplace
 I  IMS/OPER      IMS-TSO Operator Interface
 L  Loco          Logical Console Operator Interface
 N  Network Print Network Print Facility
 M  More          Additional IBM Products

      Enter X to Terminate using log/list defaults

 4B                  :01.7                                        04/14
```

After pressing the “ENTER” key the following LOCO TSO interface screen will be displayed.

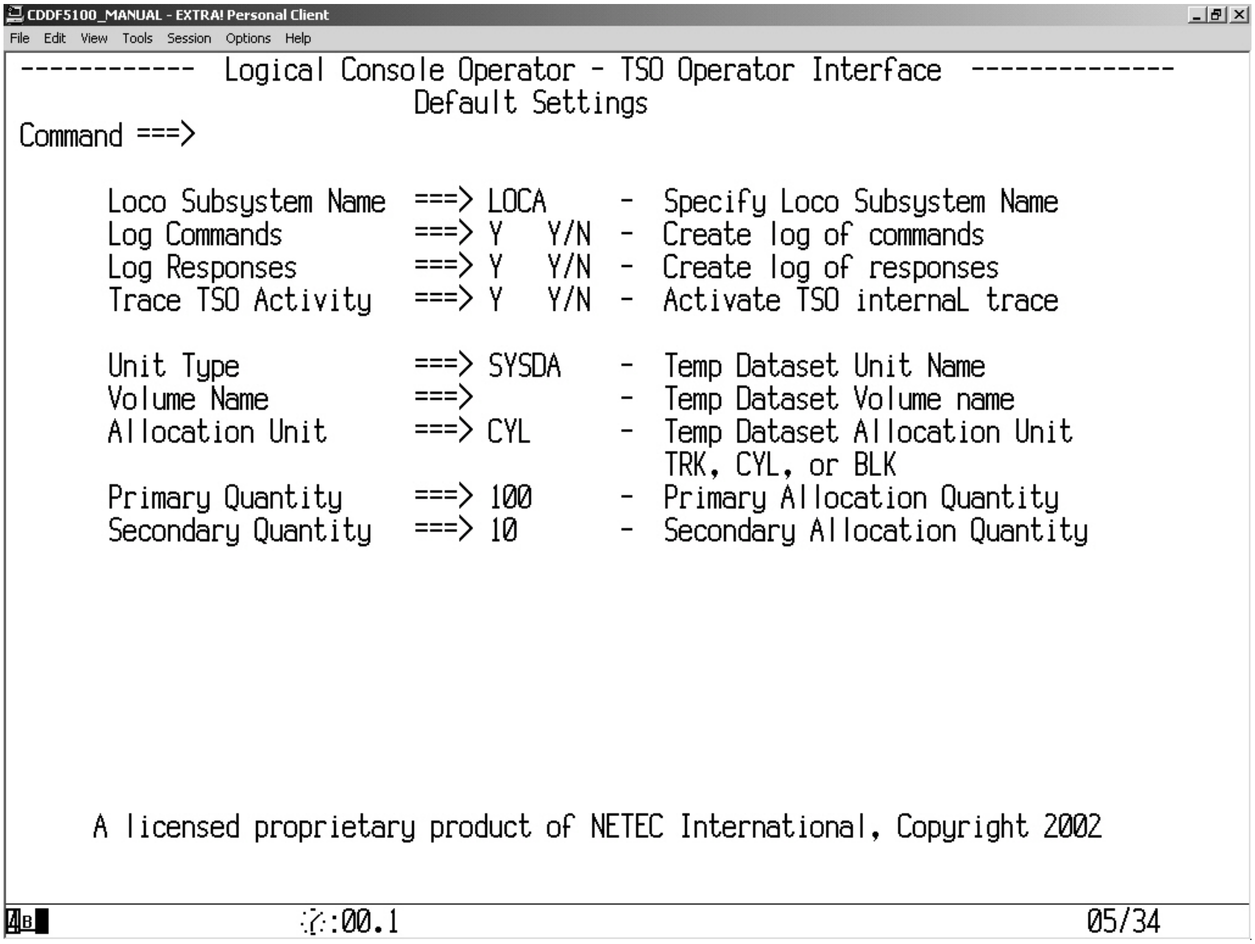


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
 ------------  Logical Console Operator - TSO Operator Interface  --------------
                            Main Selection Menu

 Command ===> _

     0    Set Operation Parameters                           USERID    - U2TEC8
     1    Display Loco Status                                TERMINAL  - 3278
     2    Perform Loco Table Maintenance                     TIME      - 10:35
     3    Manipulate Loco Trace                              DATE      - 02/10/19
     4    Activate/Deactivate/Hold/Release Resources         LOCO Name - LOCA
     5    Submit Batch Jobs Through Loco
     6    Display / Manipulate Loco Queues

      A licensed proprietary product of NETEC International, Copyright 2002
```

Setting Operational Parameters

One of the first tasks a new user of the LOCO TSO interface must do is to set the default operation parameters. The session parameters will be retained across TSO sessions. This is done by entering option 0 in the main selection menu and pressing the "ENTER" key. The following screen will be displayed.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
4B
:00.1
05/34

```
------------  Logical Console Operator - TSO Operator Interface  --------------
                          Default Settings
Command ===>

      Loco Subsystem Name  ===> LOCA     -  Specify Loco Subsystem Name
      Log Commands         ===> Y   Y/N  -  Create log of commands
      Log Responses        ===> Y   Y/N  -  Create log of responses
      Trace TSO Activity   ===> Y   Y/N  -  Activate TSO internaL trace

      Unit Type            ===> SYSDA    -  Temp Dataset Unit Name
      Volume Name          ===>          -  Temp Dataset Volume name
      Allocation Unit      ===> CYL      -  Temp Dataset Allocation Unit
                                            TRK, CYL, or BLK
      Primary Quantity     ===> 100      -  Primary Allocation Quantity
      Secondary Quantity   ===> 10       -  Secondary Allocation Quantity




     A licensed proprietary product of NETEC International, Copyright 2002
```

This screen is used to specify the various options that are to be used when executing the LOCO TSO interface. The parameter descriptions follow.

Loco Subsystem Name

This is the one to four character name of the LOCO subsystem that the user wishes to control.

Log Commands

This flag indicates whether or not the LOCO TSO interface is to log commands this user issues to the ISPF log.

Log Responses

This flag indicates whether or not the LOCO TSO interface is to log responses to commands this user issues to the ISPF log.

Trace TSO Activity

This flag indicates whether or not to create a NETEC International internal trace of the LOCO TSO interface. NETEC International technical support personnel may ask that this flag be set to Y to assist in problem resolution.

The following parameters are used to allocate the temporary dataset create as part of the LOCO table compilation process.

Unit Type

This specifies the DASD unit type to be used when dynamically allocating the LOCO table compilation output data set.

Volume Name

This specifies the volume serial number to be used when dynamically allocating the LOCO table compilation output data set. This parameter is optional.

Allocation Unit

This specifies the allocation unit to be used when dynamically allocating the LOCO table compilation output data set. The valid options are TRK, CYL, or BLK.

Primary Quantity

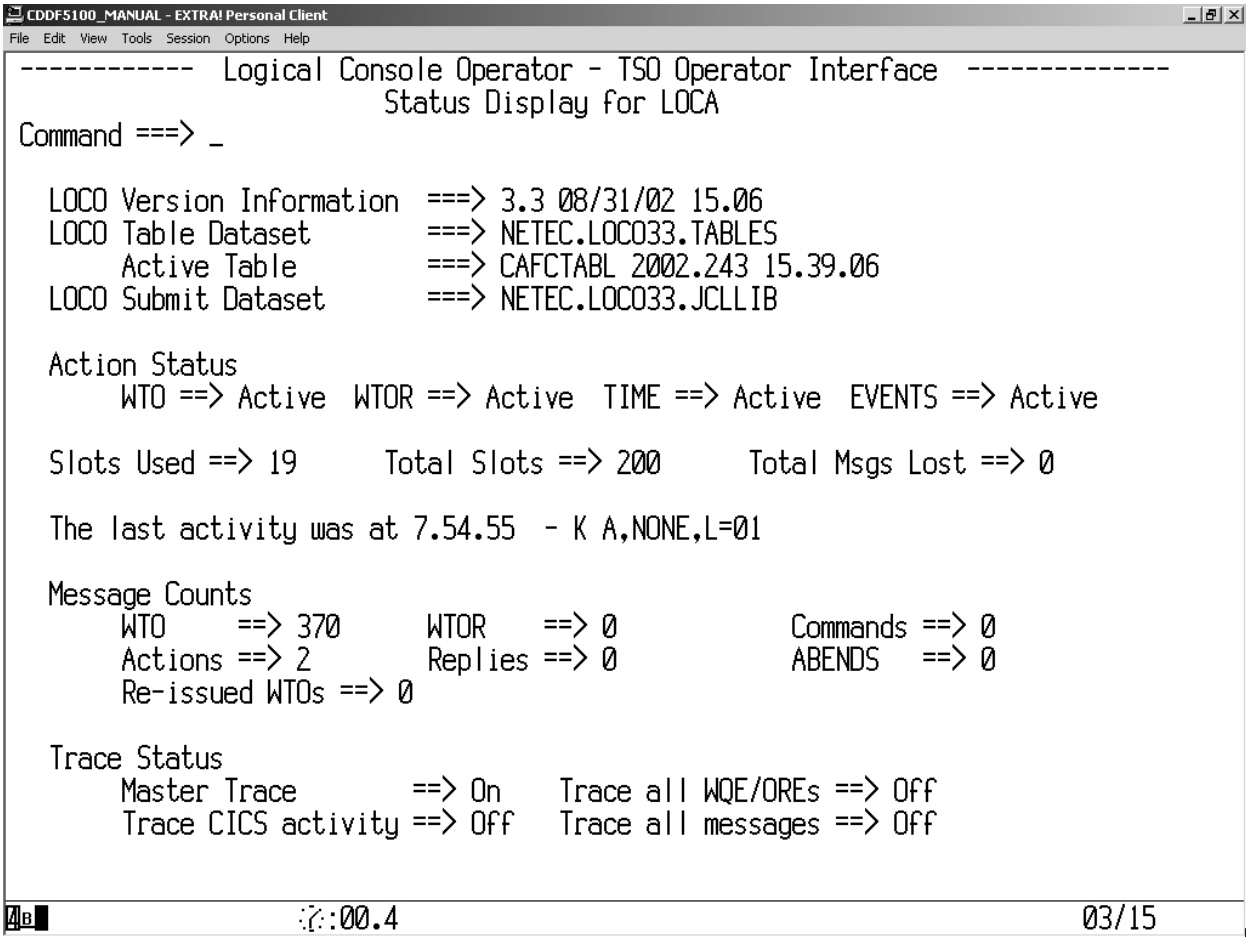
This specifies the primary quantity of the allocation unit to be used when dynamically allocating the LOCO table compilation output data set.

Secondary Quantity

This specifies the secondary quantity of the allocation unit to be used when dynamically allocating the LOCO table compilation output data set.

Displaying LOCO Status

Entering option 1 on the main selection menu and pressing the “ENTER” key displays the current status of the LOCO system. The following screen will be displayed.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
------------  Logical Console Operator - TSO Operator Interface  --------------
                       Status Display for LOCA
Command ===> _

  LOCO Version Information  ===> 3.3 08/31/02 15.06
  LOCO Table Dataset        ===> NETEC.LOCO33.TABLES
       Active Table         ===> CAFCTABL 2002.243 15.39.06
  LOCO Submit Dataset       ===> NETEC.LOCO33.JCLLIB

  Action Status
       WTO ==> Active  WTOR ==> Active  TIME ==> Active  EVENTS ==> Active

  Slots Used ==> 19      Total Slots ==> 200      Total Msgs Lost ==> 0

  The last activity was at 7.54.55  - K A,NONE,L=01

  Message Counts
       WTO     ==> 370      WTOR    ==> 0            Commands ==> 0
       Actions ==> 2        Replies ==> 0            ABENDS   ==> 0
       Re-issued WTOs ==> 0

  Trace Status
       Master Trace        ==> On    Trace all WQE/OREs ==> Off
       Trace CICS activity ==> Off   Trace all messages ==> Off
```

This is the status screen for the selected LOCO subsystem. This screen displays current vital statistics about the active LOCO subsystem.

The following describes the fields on the LOCO status display.

LOCO Ver Information ==> This information represents the current LOCO version, release date and time.
LOCO Table Dataset ==> This is the dataset LOCO will load the LOCO tables from.
Active Table ==> This is the currently active LOCO table.

LOCO Submit Dataset ==> This is the dataset LOCO will use to submit batch jobs from.

Action Status This section indicates whether actions from a queue are ACTIVE or HELD. If the actions are held, LOCO actions will be bypassed if a triggering event occurs.

WTO ==> Status of WTO actions

WTOR ==> Status of WTOR actions

TIME ==> Status of Time actions

EVENTS ==> Status of Event actions

Slots Status This section contains the status of storage reserved for LOCO to save messages away for processing. Each slot represents one message.

Slots Used ==> This value represents the high water mark of slots used during the current execution of the LOCO subsystem.

Total Slots ==> This value is the total number of slots defined for this execution of LOCO. This value is specified as the QLIMIT parameter on the DEFINE card.

Total Msgs Lost ==> This value represents the total number of message LOCO could not process due to exceeding the QLIMIT count on the DEFINE card.

The last activity was This field contains the lst command that LOCO issued and the date and time it was issued.

Message Counts These fields represent the message activity LOCO has encountered during this execution.

WTO ==> This value is the number of WTOs intercepted.

WTOR == This value is the number of WTORs intercepted.

Commands ==> This value is the number of commands issued.

Actions ==> This value is the number of LOCO actions taken.

Replies ==> This value is the number of replies issued by LOCO.

ABENDS ==> This value is the number of abends detected.

Re-issued WTOs ==> This value is the number of times LOCO re-issued a WTO.

```
Trace Status                 The following fields display the
                             current status of the LOCO internal
                             traces.
    Master Trace         ==> This field indicates whether the
                             master internal is active.
    Trace all WQE/OREs   ==> This field indicates whether LOCO is
                             to trace all WQE/ORE activity.
    Trace CICS activity ==> This field indicates whether LOCO is
                             to trace all CICS activity.
    Trace all messages   ==> This field indicates whether LOCO is
                             to trace all message activity.
```

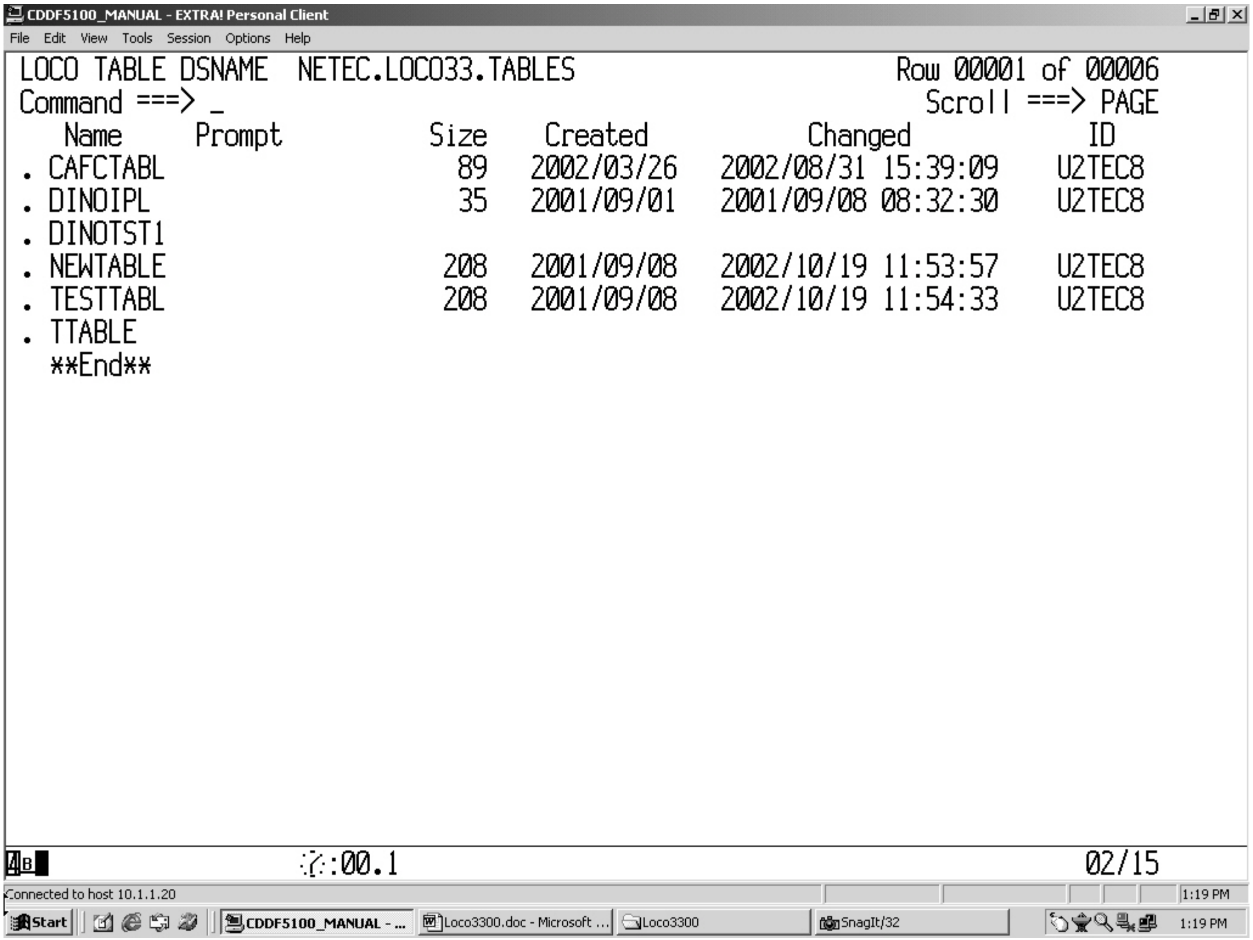
Security Requirments

To issue the LOCO status command requires READ access to the following SAF Profiles.

```
     LOCO.xxxx.SSI    - to access to LOCO subsystem.
     LOCO.xxxx.STATUS - to allow display access.

     xxxx  =  The one to four character LOCO subsytem name
```

Performing LOCO Table Maintenance

Entering option 2 on the main selection menu and pressing the "ENTER" key displays the LOCO Table Maintenance menu. The following screen will be displayed. This screen contains a member list of all the LOCO tables current present in the LOCO table library being used by the LOCO subsystem specified under operational parameters.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
LOCO TABLE DSNAME  NETEC.LOCO33.TABLES                    Row 00001 of 00006
Command ===> _                                               Scroll ===> PAGE
   Name     Prompt          Size   Created        Changed           ID
. CAFCTABL                    89  2002/03/26  2002/08/31 15:39:09   U2TEC8
. DINOIPL                     35  2001/09/01  2001/09/08 08:32:30   U2TEC8
. DINOTST1
. NEWTABLE                   208  2001/09/08  2002/10/19 11:53:57   U2TEC8
. TESTTABL                   208  2001/09/08  2002/10/19 11:54:33   U2TEC8
. TTABLE
  **End**
```

:00.1
02/15
Connected to host 10.1.1.20
1:19 PM
Start
CDDF5100_MANUAL - ...
Loco3300.doc - Microsoft ...
Loco3300
SnagIt/32
1:19 PM

The screen is the member selection screen for the LOCO table library. The list contains all LOCO table members currently available for browsing, editing or loading to batch through LOCO. The user can enter three actions in the field preceeding the member name.
The user can:

1. Request LOCO load the table for processing.
2. Broswe the LOCO table member.
3. Modify the member contents through ISFP edit.

After successful modification, the LOCOBLDR process will be involked to rebuild the LOCO table and the user will have the ability to replace the exiting LOCO table in the LOCO table library.

The user indicates the required action by placing an:
L in the action field to request LOCO load the table.
B in the action field to browse the member.
E in the action field to edit the member.

Standard ISPF statistics are maintained for the LOCO Table library.

Security Requirments

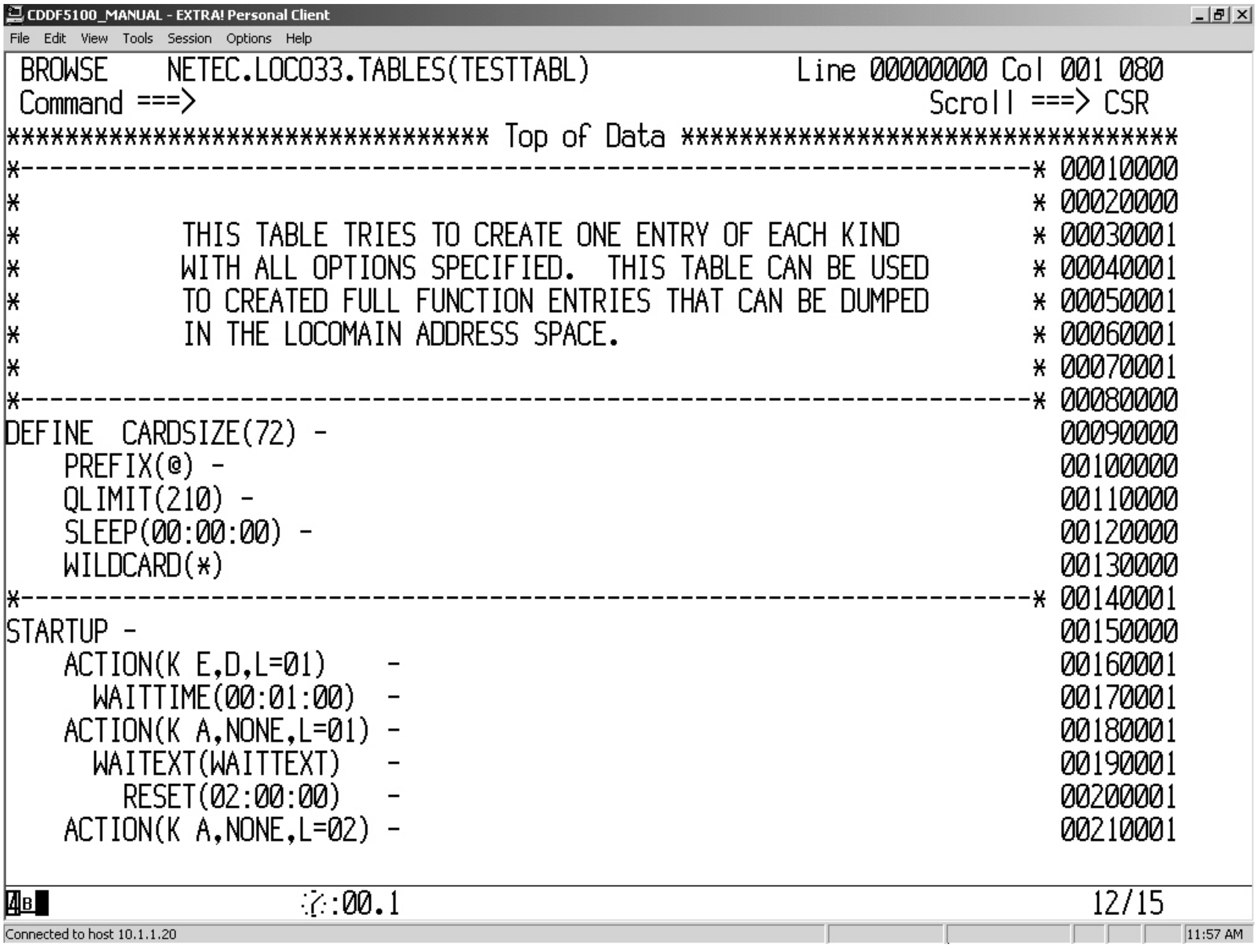
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI - to access the LOCO subsystem.
LOCO.xxxx.TABLES.BROWSE - to display the LOCO table entries.
LOCO.xxxx.TABLES.EDIT - to edit the LOCO table entries.
LOCO.xxxx.TABLES.LOAD - to request LOCO load the table entry.

xxxx = The one to four character LOCO subsytem name

Browsing a LOCO Table

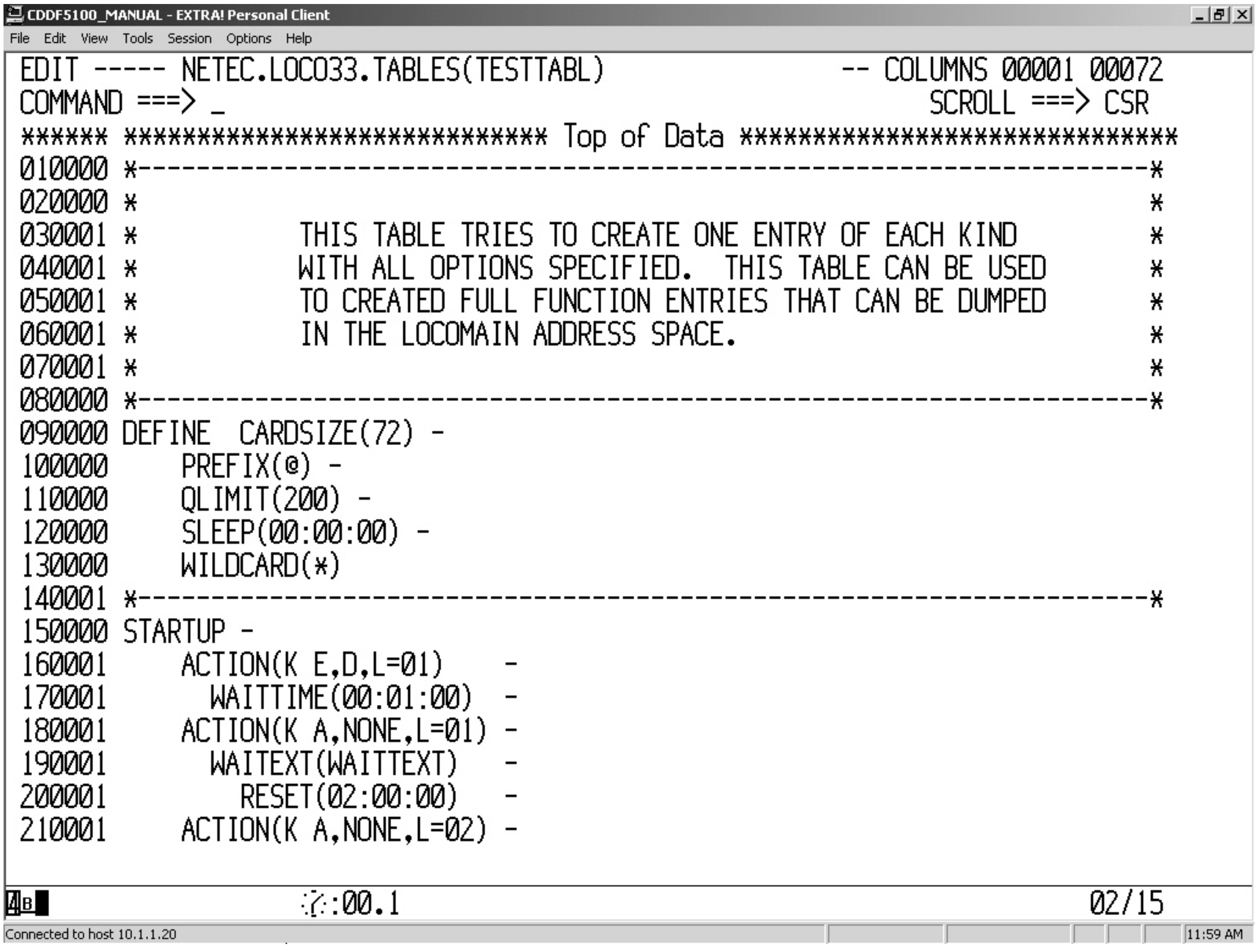
The following screen will be displayed when a “B” is entered in front of a member name in the manipulate LOCO tables member name list. The LOCO table will be displayed using standard ISPF browse. Normal ISPF browse keys are active during this session. The user can scroll up and down browsing the selected table. The “END” or “CANCEL” command will terminate the browse session and return to the manipulate LOCO Tables member list screen.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
 BROWSE    NETEC.LOCO33.TABLES(TESTTABL)           Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> CSR
********************************* Top of Data **********************************
*-----------------------------------------------------------------------* 00010000
*                                                                       * 00020000
*          THIS TABLE TRIES TO CREATE ONE ENTRY OF EACH KIND            * 00030001
*          WITH ALL OPTIONS SPECIFIED.  THIS TABLE CAN BE USED          * 00040001
*          TO CREATED FULL FUNCTION ENTRIES THAT CAN BE DUMPED          * 00050001
*          IN THE LOCOMAIN ADDRESS SPACE.                               * 00060001
*                                                                       * 00070001
*-----------------------------------------------------------------------* 00080000
DEFINE  CARDSIZE(72) -                                                    00090000
    PREFIX(@) -                                                           00100000
    QLIMIT(210) -                                                         00110000
    SLEEP(00:00:00) -                                                     00120000
    WILDCARD(*)                                                           00130000
*-----------------------------------------------------------------------* 00140001
STARTUP -                                                                 00150000
    ACTION(K E,D,L=01)    -                                               00160001
      WAITTIME(00:01:00)  -                                               00170001
    ACTION(K A,NONE,L=01) -                                               00180001
      WAITEXT(WAITTEXT)   -                                               00190001
        RESET(02:00:00)   -                                               00200001
    ACTION(K A,NONE,L=02) -                                               00210001

4B                                    :00.1                          12/15
```

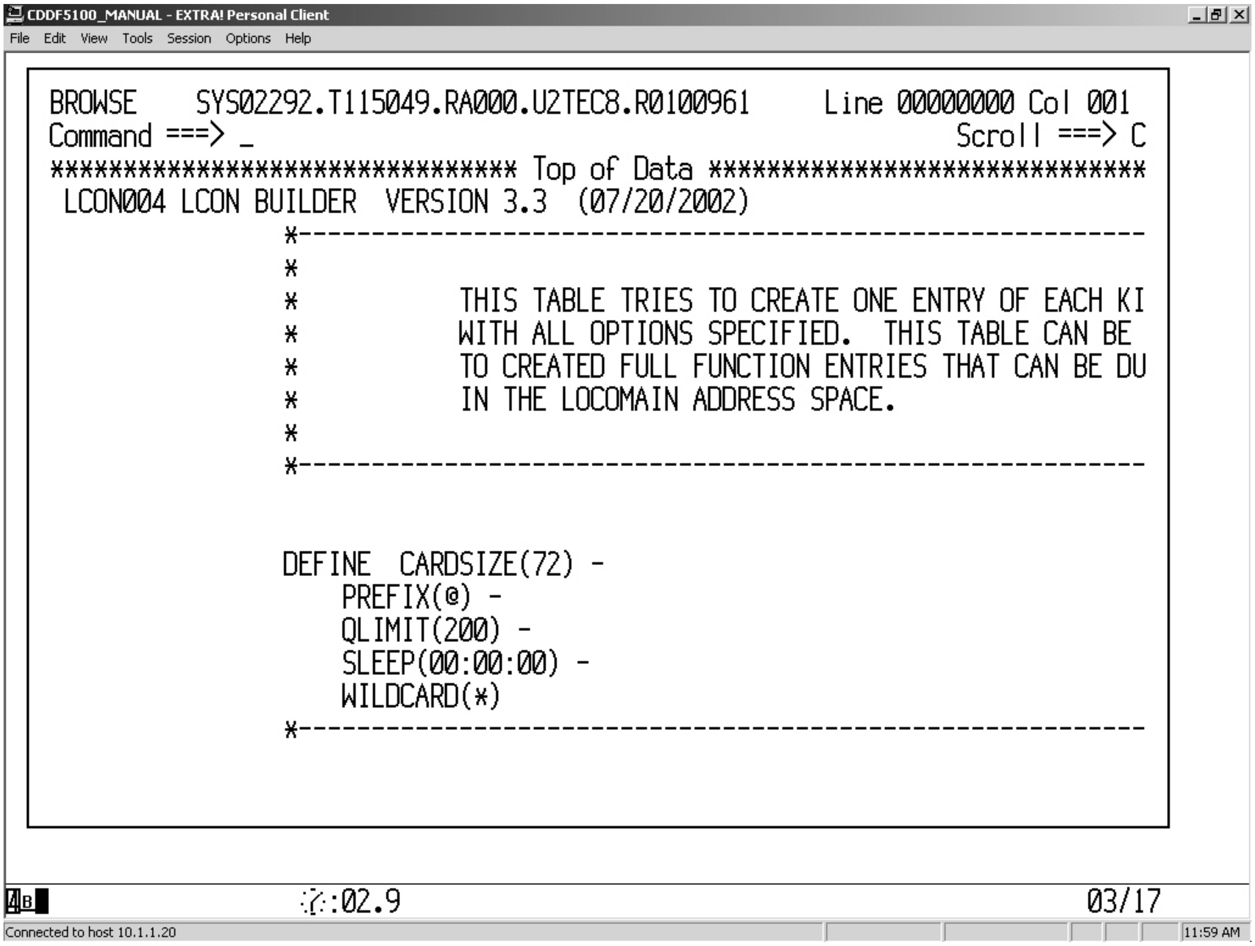
Editing a LOCO Table

The following screen will be displayed when a “E” is entered in front of a member name in the manipulate LOCO tables member name list. The LOCO table will be displayed using standard ISPF editor. Normal ISPF edit keys are active during this session. The user can scroll up and down in the selected table making modifications as needed. If the user has modified the table and entered the “END” command, the LOCO compilation process will be automatically invoked. The “CANCEL” command will terminate the edit session without saving any table modifications and return to the manipulate LOCO Tables member list screen.


CDDF5100_MANUAL - EXTRA! Personal Client
File
Edit
View
Tools
Session
Options
Help
Connected to host 10.1.1.20
11:59 AM

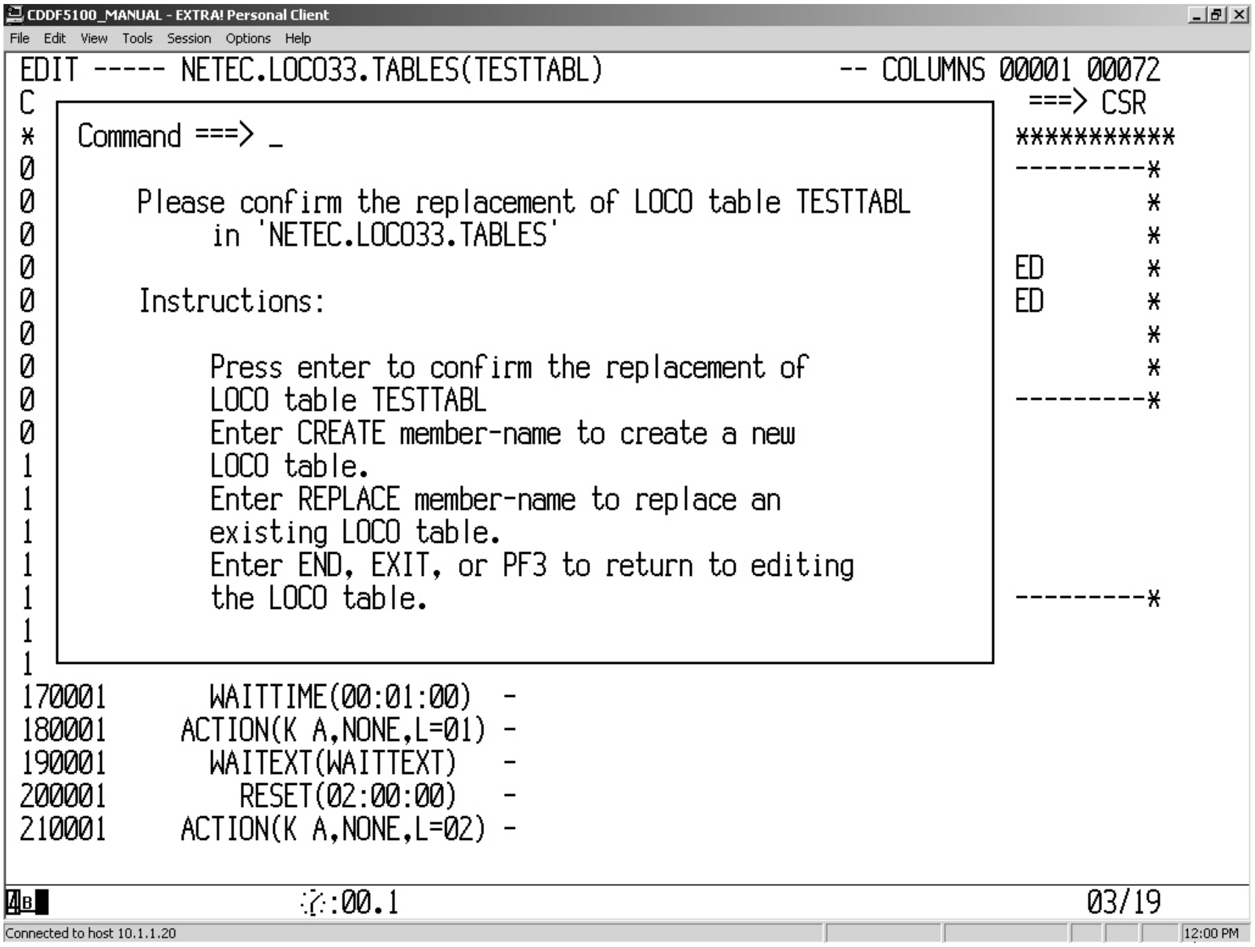
```
EDIT ----- NETEC.LOCO33.TABLES(TESTTABL)                -- COLUMNS 00001 00072
COMMAND ===> _                                                 SCROLL ===> CSR
****** ***************************** Top of Data ******************************
010000 *-----------------------------------------------------------------------*
020000 *                                                                       *
030001 *           THIS TABLE TRIES TO CREATE ONE ENTRY OF EACH KIND           *
040001 *           WITH ALL OPTIONS SPECIFIED.  THIS TABLE CAN BE USED         *
050001 *           TO CREATED FULL FUNCTION ENTRIES THAT CAN BE DUMPED         *
060001 *           IN THE LOCOMAIN ADDRESS SPACE.                              *
070001 *                                                                       *
080000 *-----------------------------------------------------------------------*
090000 DEFINE  CARDSIZE(72) -
100000     PREFIX(@) -
110000     QLIMIT(200) -
120000     SLEEP(00:00:00) -
130000     WILDCARD(*)
140001 *-----------------------------------------------------------------------*
150000 STARTUP -
160001     ACTION(K E,D,L=01)    -
170001       WAITTIME(00:01:00)  -
180001     ACTION(K A,NONE,L=01) -
190001       WAITEXT(WAITTEXT)   -
200001         RESET(02:00:00)   -
210001     ACTION(K A,NONE,L=02) -

4B                    :00.1                                               02/15
```

If the user has modified the table and has entered the “END” command or presses PF3 the LOCO compilation process will be automatically started. At the completion of the LOCO table compile, the output from the compile process will be made available to the user through standard ISPF browse. The user can scroll through the output and determine if the compile was successful. The following is an example of the output of the compile process.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
BROWSE    SYS02292.T115049.RA000.U2TEC8.R0100961     Line 00000000 Col 001
Command ===> _                                                Scroll ===> C
******************************** Top of Data ******************************
 LCON004 LCON BUILDER  VERSION 3.3  (07/20/2002)
                *-----------------------------------------------------------
                *
                *           THIS TABLE TRIES TO CREATE ONE ENTRY OF EACH KI
                *           WITH ALL OPTIONS SPECIFIED.  THIS TABLE CAN BE
                *           TO CREATED FULL FUNCTION ENTRIES THAT CAN BE DU
                *           IN THE LOCOMAIN ADDRESS SPACE.
                *
                *-----------------------------------------------------------


                DEFINE  CARDSIZE(72) -
                    PREFIX(@) -
                    QLIMIT(200) -
                    SLEEP(00:00:00) -
                    WILDCARD(*)
                *-----------------------------------------------------------
```

If the user enters the “END” command or presses PF3 and the table compile was successful the user will be prompted with the following popup panel.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
EDIT ----- NETEC.LOCO33.TABLES(TESTTABL)            -- COLUMNS 00001 00072
C                                                          ===> CSR
*   Command ===> _                                       ***********
0                                                        ---------*
0      Please confirm the replacement of LOCO table TESTTABL       *
0           in 'NETEC.LOCO33.TABLES'                               *
0                                                        ED        *
0      Instructions:                                     ED        *
0                                                                  *
0           Press enter to confirm the replacement of              *
0           LOCO table TESTTABL                          ---------*
0           Enter CREATE member-name to create a new
1           LOCO table.
1           Enter REPLACE member-name to replace an
1           existing LOCO table.
1           Enter END, EXIT, or PF3 to return to editing
1           the LOCO table.                              ---------*
1
1
170001        WAITTIME(00:01:00)  -
180001      ACTION(K A,NONE,L=01) -
190001        WAITEXT(WAITTEXT)   -
200001          RESET(02:00:00)   -
210001      ACTION(K A,NONE,L=02) -
```

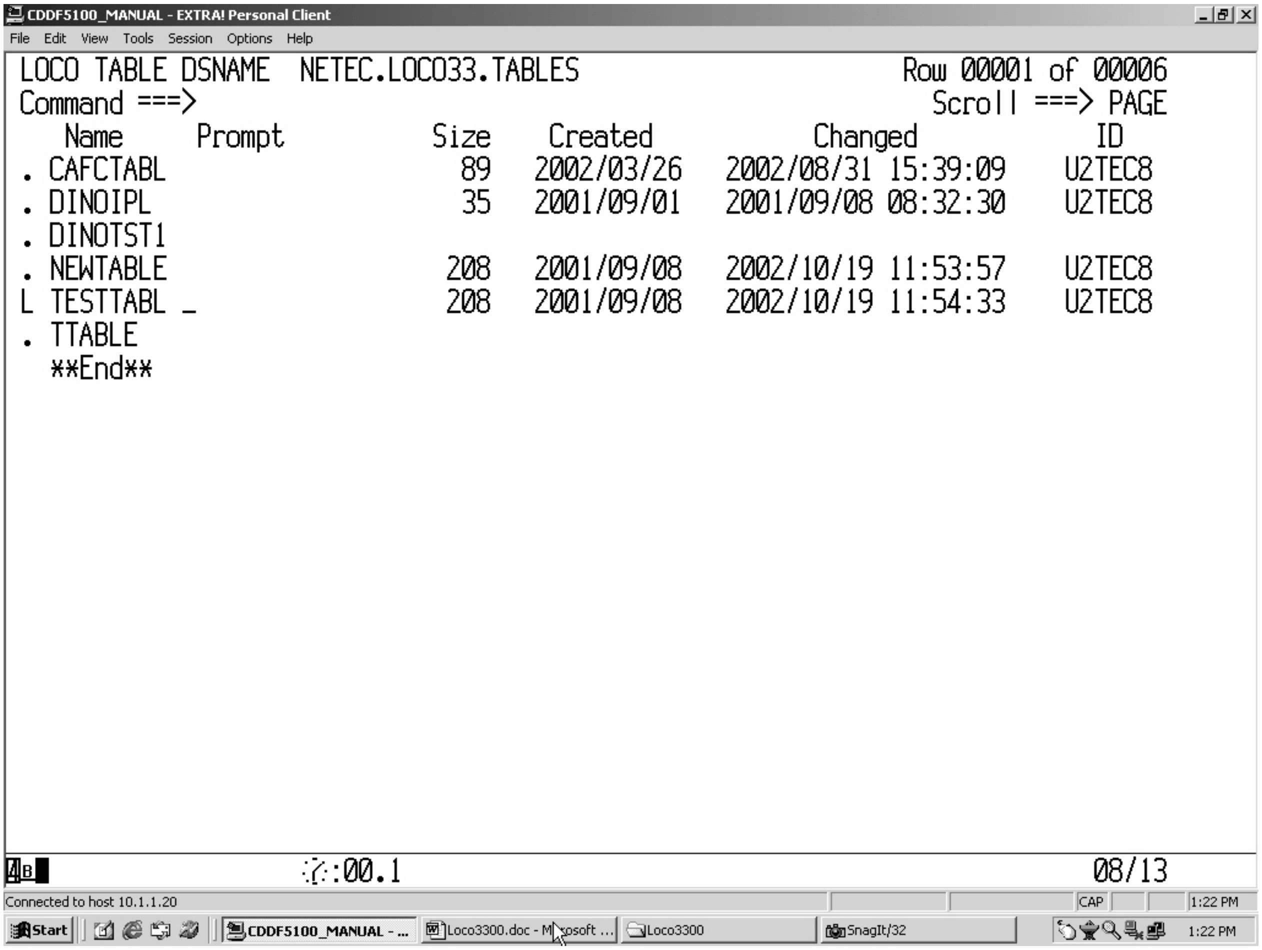
The user can specify one of four options from this menu. The user can either:

1. Press “ENTER” to confirm the replacement of the table that has just been edited and recompiled.
2. Enter the “CREATE” command to have a new table member created in the LOCO table library.
3. Enter the “REPLACE” command to replace an existing table member in the LOCO table library.
4. Enter “END”, “EXIT”, or “PF3” to terminate the table edit without saving the modified table

The ISPF statistics will be updated as part of the table save process.

Loading a LOCO Table

By entering an “L” in front of a member name in the manipulate LOCO tables member name list and pressing enter a new copy of the selected LOCO table will be loaded by the LOCO subsystem. The following example shows the LOCO table TESTTABL being reloaded.

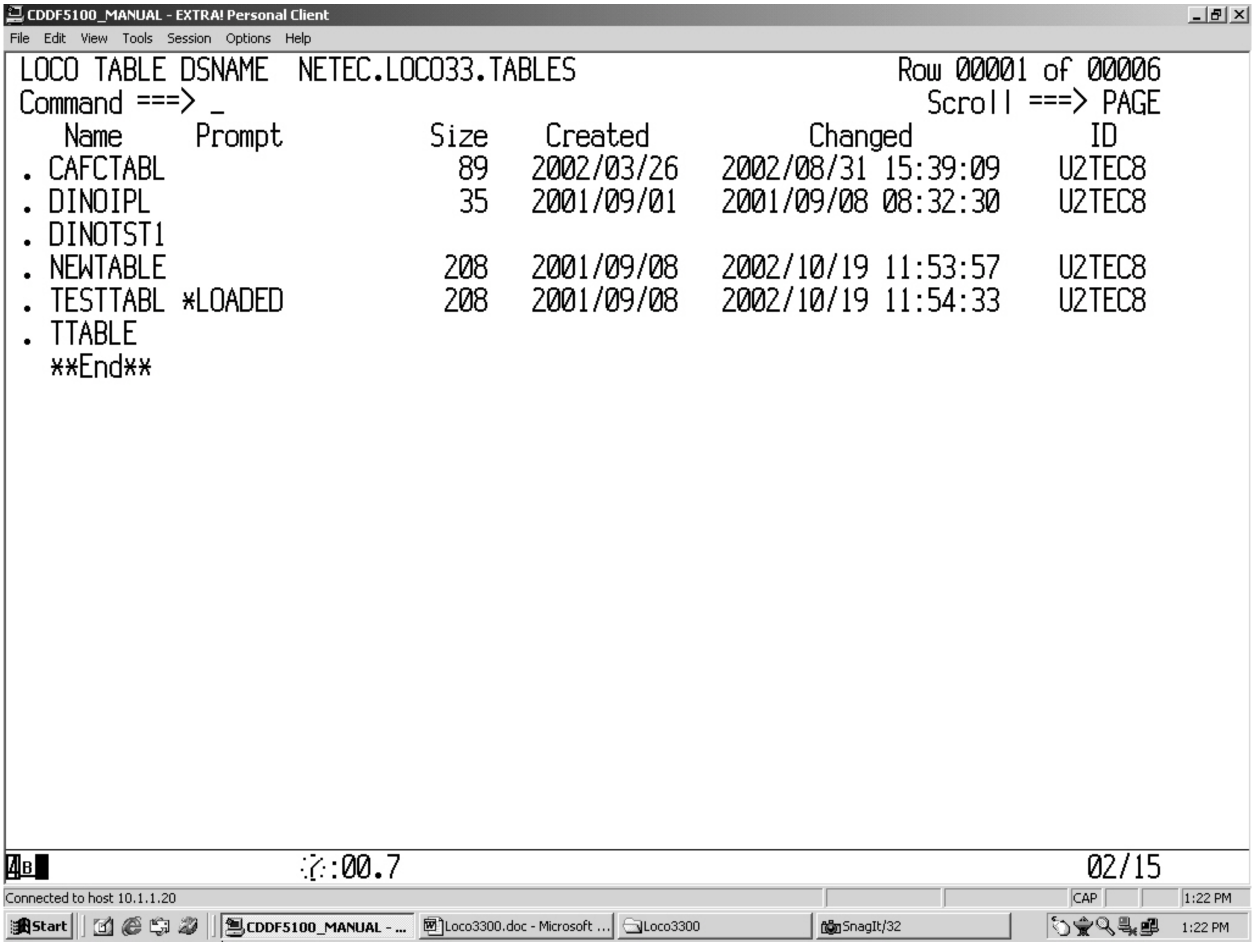


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
LOCO TABLE DSNAME  NETEC.LOCO33.TABLES                         Row 00001 of 00006
Command ===>                                                    Scroll ===> PAGE
   Name     Prompt          Size    Created        Changed            ID
. CAFCTABL                    89   2002/03/26   2002/08/31 15:39:09   U2TEC8
. DINOIPL                     35   2001/09/01   2001/09/08 08:32:30   U2TEC8
. DINOTST1
. NEWTABLE                   208   2001/09/08   2002/10/19 11:53:57   U2TEC8
L TESTTABL _                 208   2001/09/08   2002/10/19 11:54:33   U2TEC8
. TTABLE
  **End**

4B                     :00.1                                          08/13
```

Connected to host 10.1.1.20
CAP
1:22 PM
Start
CDDF5100_MANUAL - ...
Loco3300.doc - Microsoft ...
Loco3300
SnagIt/32
1:22 PM

Successful reloading of the LOCO table will be indicated in the LOCO Table member list as illustrated below.



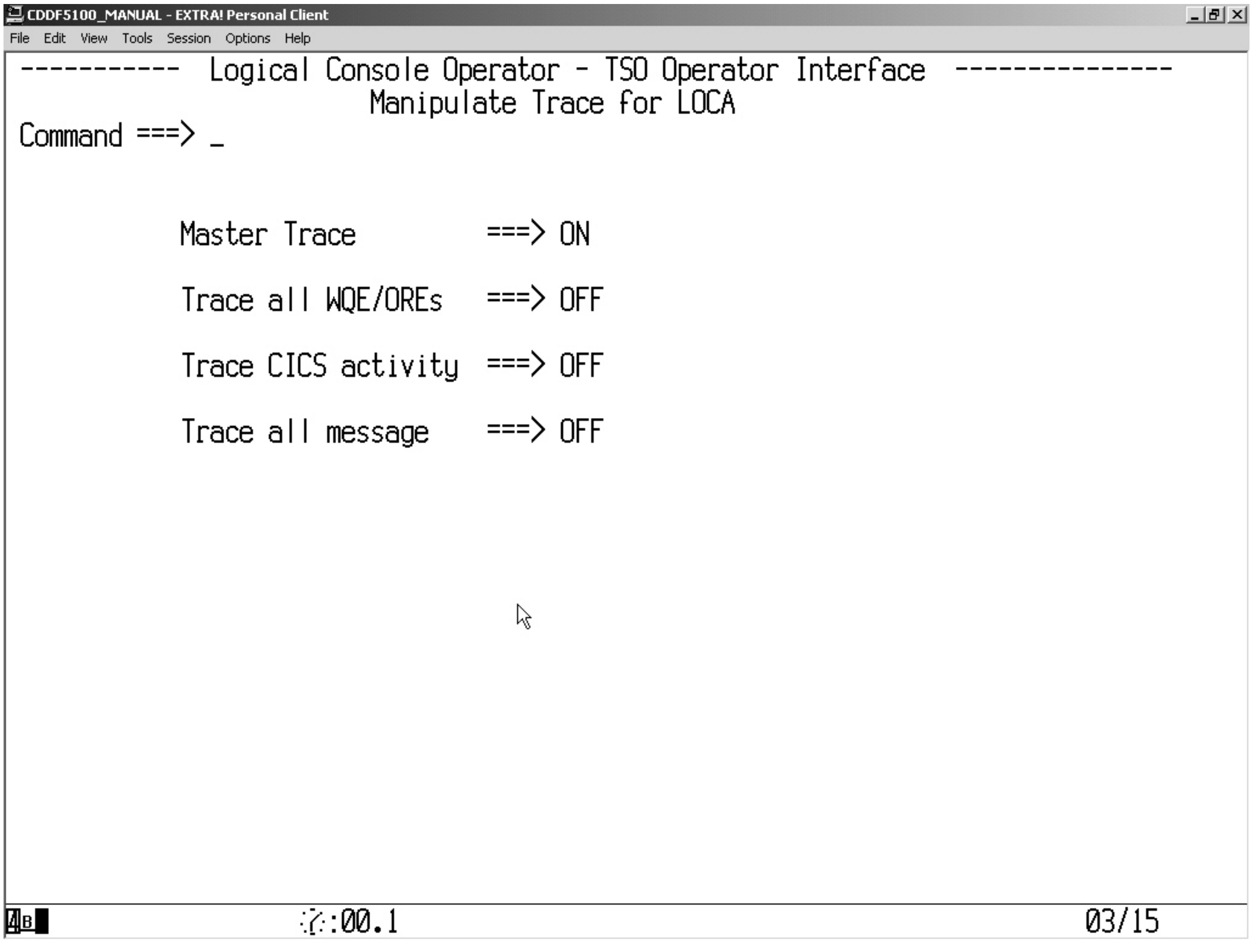
CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
LOCO TABLE DSNAME  NETEC.LOCO33.TABLES                      Row 00001 of 00006
Command ===> _                                               Scroll ===> PAGE
   Name     Prompt          Size   Created        Changed            ID
. CAFCTABL                    89   2002/03/26   2002/08/31 15:39:09   U2TEC8
. DINOIPL                     35   2001/09/01   2001/09/08 08:32:30   U2TEC8
. DINOTST1
. NEWTABLE                   208   2001/09/08   2002/10/19 11:53:57   U2TEC8
. TESTTABL *LOADED           208   2001/09/08   2002/10/19 11:54:33   U2TEC8
. TTABLE
  **End**
```

4B
:00.7
02/15
Connected to host 10.1.1.20
CAP
1:22 PM
Start
CDDF5100_MANUAL - ...
Loco3300.doc - Microsoft ...
Loco3300
SnagIt/32
1:22 PM

Manipulating LOCO Trace

The following screen is displayed when option "2" is entered from the selection menu. The screen allows the user to start and stop LOCO internal traces.



CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
-----------  Logical Console Operator - TSO Operator Interface  ---------------
                        Manipulate Trace for LOCA
Command ===> _


           Master Trace         ===> ON

           Trace all WQE/OREs   ===> OFF

           Trace CICS activity  ===> OFF

           Trace all message    ===> OFF















4B                    :00.1                                              03/15
```

Trace status is changed by keying ON or OFF into the field next to the type of trace field.

The following describes the types of internal traces the user can control.

```
  Master Trace         ===> This field instructs LOCO to either
                             activate or deactivate the master
                             internal trace.
  Trace all WQE/OREs   ===> This field instructs LOCO to either
                             activate or deactivate the tracing of
                             all WQEs/OREs.
```

Trace CICS activity ===> This field instructs LOCO to either activate or deactivate the tracing of CICS activity.
Trace all message ===> This field instructs LOCO to either activate or deactivate the tracing of all messages.

Security Requirments

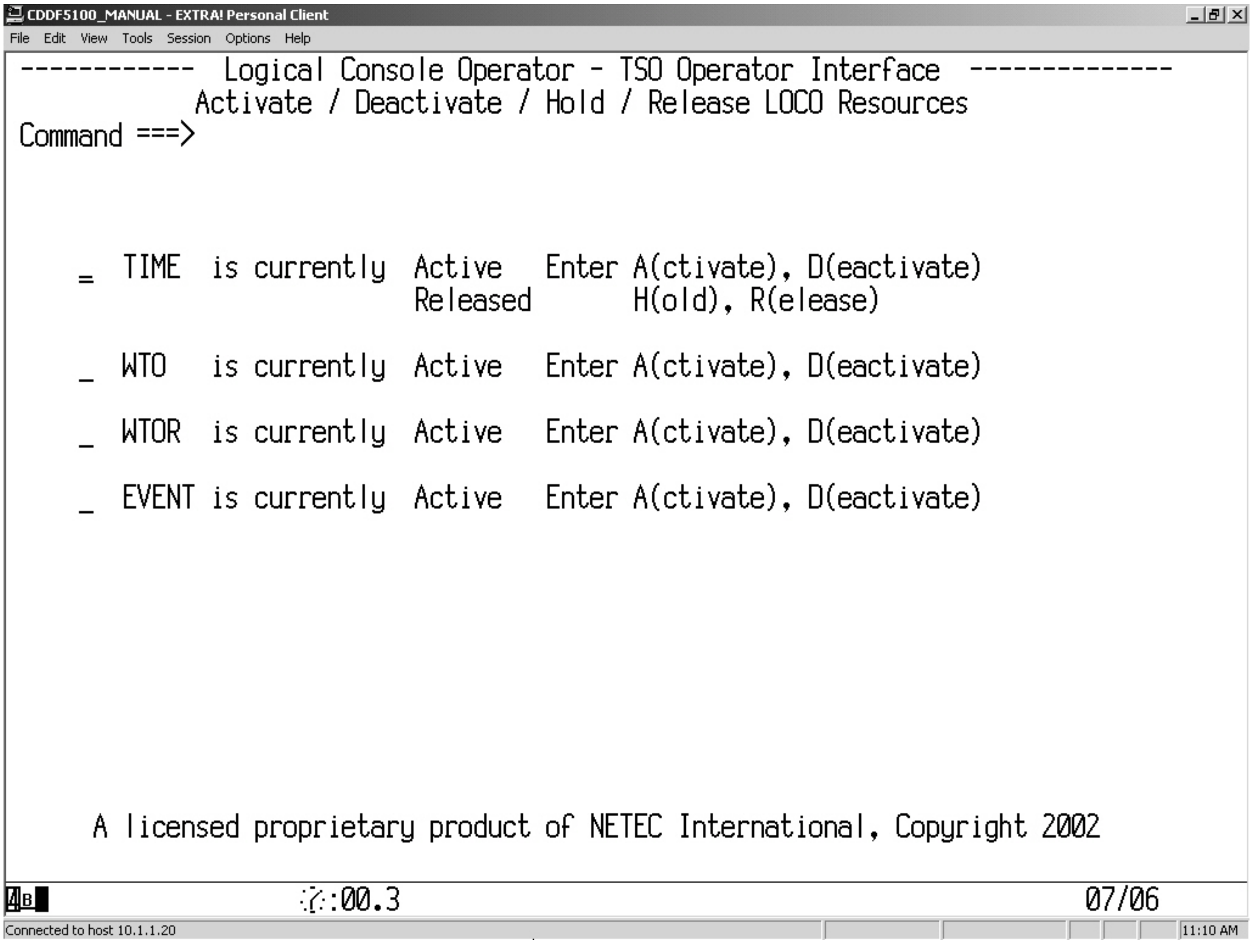
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI - to access to LOCO subsystem.
LOCO.xxxx.TRACES - to allow access to change traces.

xxxx = The one to four character LOCO subsytem name

Activating, Deactivating, Holding, and Releasing Resources

The following screen is displayed when option “4” is entered from the selection menu. The screen allows the user to start and stop LOCO internal traces.



CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
------------  Logical Console Operator - TSO Operator Interface  --------------
              Activate / Deactivate / Hold / Release LOCO Resources
Command ===>



   =  TIME  is currently  Active   Enter A(ctivate), D(eactivate)
                          Released       H(old), R(elease)

   _  WTO   is currently  Active   Enter A(ctivate), D(eactivate)

   _  WTOR  is currently  Active   Enter A(ctivate), D(eactivate)

   _  EVENT is currently  Active   Enter A(ctivate), D(eactivate)







     A licensed proprietary product of NETEC International, Copyright 2002

4B                :00.3                                              07/06
```

From this screen the user can activate, deactivate, hold, and release LOCO resources. When LOCO resources are deactivated they are ineligible to be triggered for execution. This screen gives the user the ability to manipulate entire LOCO queues.

The following describes the fields on this screen.

```
    TIME  is currently  ===> Active  -  The queue is available to
                                        be triggered for action.
                             Paused  - The queue is unavailable
                                        to be triggered for
                                        action.
                             Held    - Queue is currently held.
                             Released- Queue is currently
                                        released.
    WTO   is currently  ===> Active  - The queue is available to
                                        be triggered for action.
                             Paused  - The queue is unavailable
                                        to be triggered for
                                        action.
   WTOR  is currently  ===> Active   - The queue is available to
                                        be triggered for action.
                            Paused   - The queue is unavailable
                                        to be triggered for
                                        action.
   EVENT is currently  ===> Active   - The queue is available to
                                        be triggered for action.
                            Paused   - The queue is unavailable
                                        to be triggered for
                                        action.
```

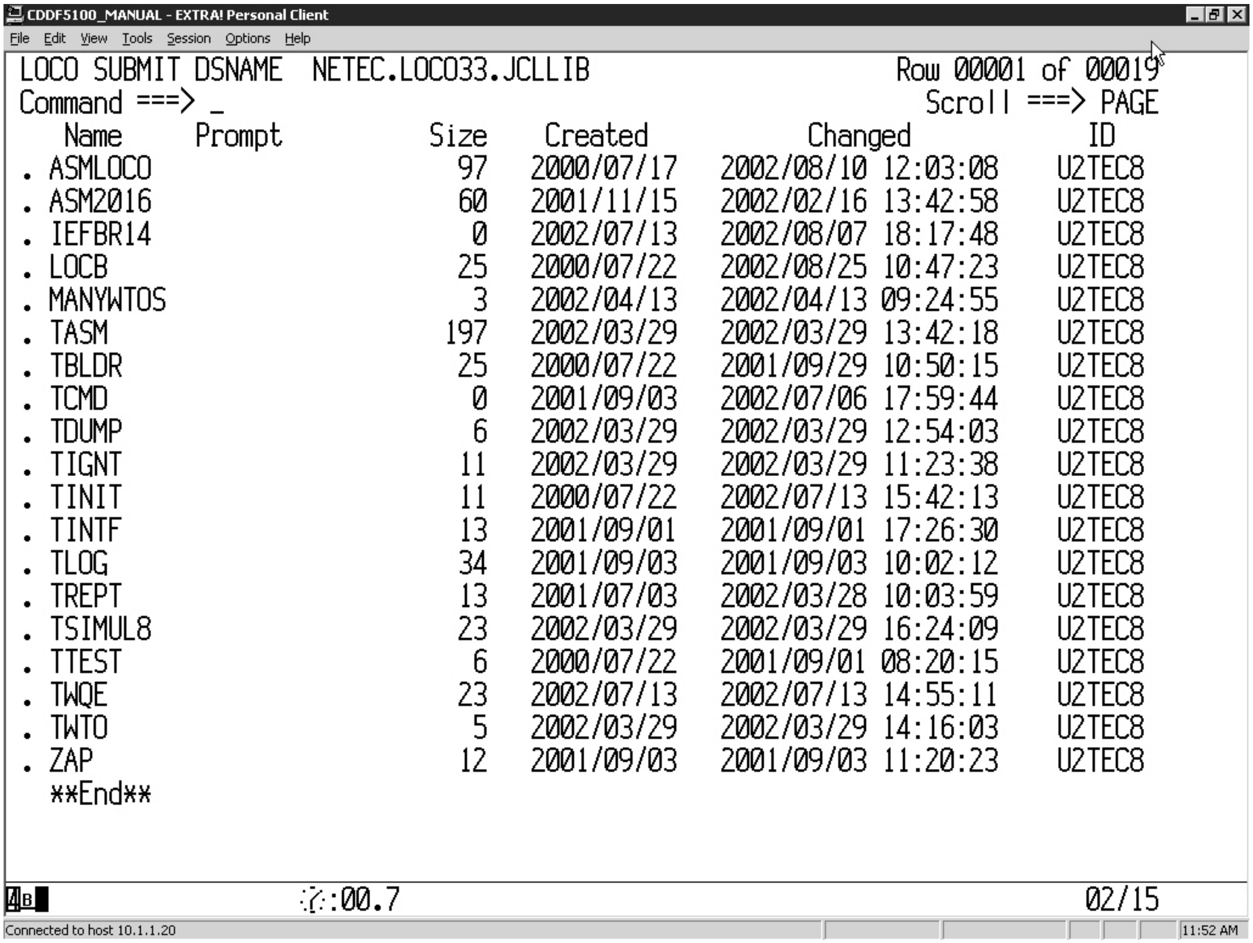
Security Requirments

To manipulate these resources requires READ access to the following SAF Profiles.

```
     LOCO.xxxx.SSI                - to access the LOCO subsystem
     LOCO.xxxx.RESOURCES.BROWSE   - to display resources
     LOCO.xxxx.RESOURCES.HOLD     - to hold/release resources
     LOCO.xxxx.RESOURCES.ACTIVATE - to activate/deactivate
                                    resources

     xxxx  =  The one to four character LOCO subsytem name
```

Submitting Batch Jobs Through LOCO

The following screen is displayed when option “5” is entered from the selection menu. The screen lets the user request jobs from the LOCO submit library be submitted for batch execution by the LOCO subsystem.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 10.1.1.20
11:52 AM

```
LOCO SUBMIT DSNAME   NETEC.LOCO33.JCLLIB                     Row 00001 of 00019
Command ===> _                                                 Scroll ===> PAGE
   Name      Prompt           Size    Created          Changed            ID
. ASMLOCO                       97   2000/07/17   2002/08/10 12:03:08   U2TEC8
. ASM2016                       60   2001/11/15   2002/02/16 13:42:58   U2TEC8
. IEFBR14                        0   2002/07/13   2002/08/07 18:17:48   U2TEC8
. LOCB                          25   2000/07/22   2002/08/25 10:47:23   U2TEC8
. MANYWTOS                       3   2002/04/13   2002/04/13 09:24:55   U2TEC8
. TASM                         197   2002/03/29   2002/03/29 13:42:18   U2TEC8
. TBLDR                         25   2000/07/22   2001/09/29 10:50:15   U2TEC8
. TCMD                           0   2001/09/03   2002/07/06 17:59:44   U2TEC8
. TDUMP                          6   2002/03/29   2002/03/29 12:54:03   U2TEC8
. TIGNT                         11   2002/03/29   2002/03/29 11:23:38   U2TEC8
. TINIT                         11   2000/07/22   2002/07/13 15:42:13   U2TEC8
. TINTF                         13   2001/09/01   2001/09/01 17:26:30   U2TEC8
. TLOG                          34   2001/09/03   2001/09/03 10:02:12   U2TEC8
. TREPT                         13   2001/07/03   2002/03/28 10:03:59   U2TEC8
. TSIMUL8                       23   2002/03/29   2002/03/29 16:24:09   U2TEC8
. TTEST                          6   2000/07/22   2001/09/01 08:20:15   U2TEC8
. TWQE                          23   2002/07/13   2002/07/13 14:55:11   U2TEC8
. TWTO                           5   2002/03/29   2002/03/29 14:16:03   U2TEC8
. ZAP                           12   2001/09/03   2001/09/03 11:20:23   U2TEC8
  **End**


4B                   :00.7                                                02/15
```

The screen is the member selection screen for the LOCO submit library. The list contains all members currently available for browsing, editing, or submitting to batch through LOCO.

The user can enter three actions in the field preceding the member name.
The user can:

1. Request LOCO submit the job for batch execution.
2. Browse the member.
3. Modify the member contents through ISFP edit.

The user indicates the required action by placing a “S” in the action field for a request to LOCO to submit the job, a “B” in the action field to browse the member of an “E” in the action field to edit the member.

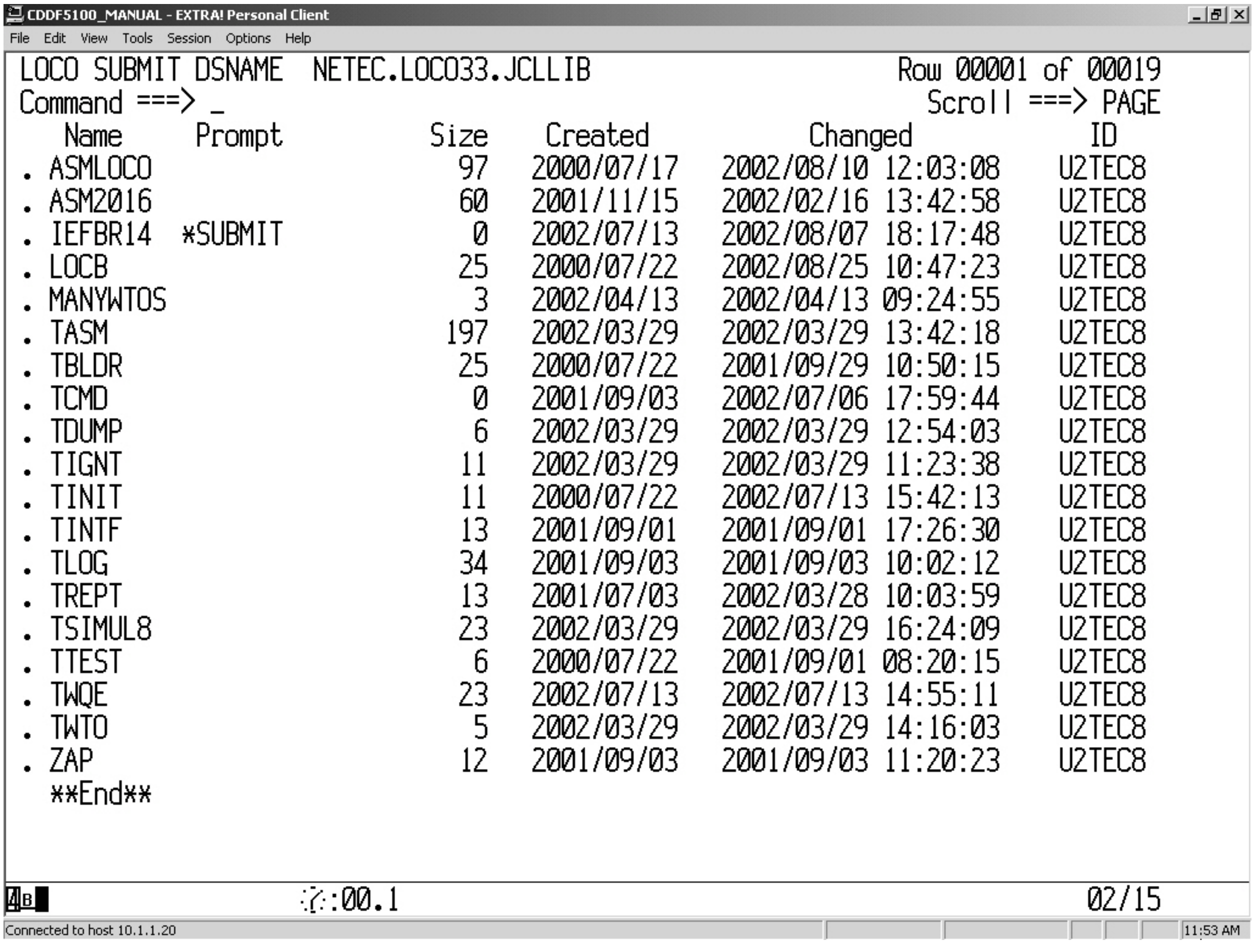
Standard ISPF statistics are maintained for the LOCO Submit library.

Security Requirments

To manipulate these resources requires READ access to the following SAF Profiles.

```
    LOCO.xxxx.SSI            - to access the LOCO subsystem.
    LOCO.xxxx.SUBMIT.BROWSE - to display the queue entries.
    LOCO.xxxx.SUBMIT.EDIT   - to edit the submit data set.
    LOCO.xxxx.SUBMIT.SUBMIT - to request LOCO submit the job.

     xxxx  =  The one to four character LOCO subsytem name
```

Successful submission of the LOCO batch job will be indicated in the LOCO submit library member list as illustrated below.



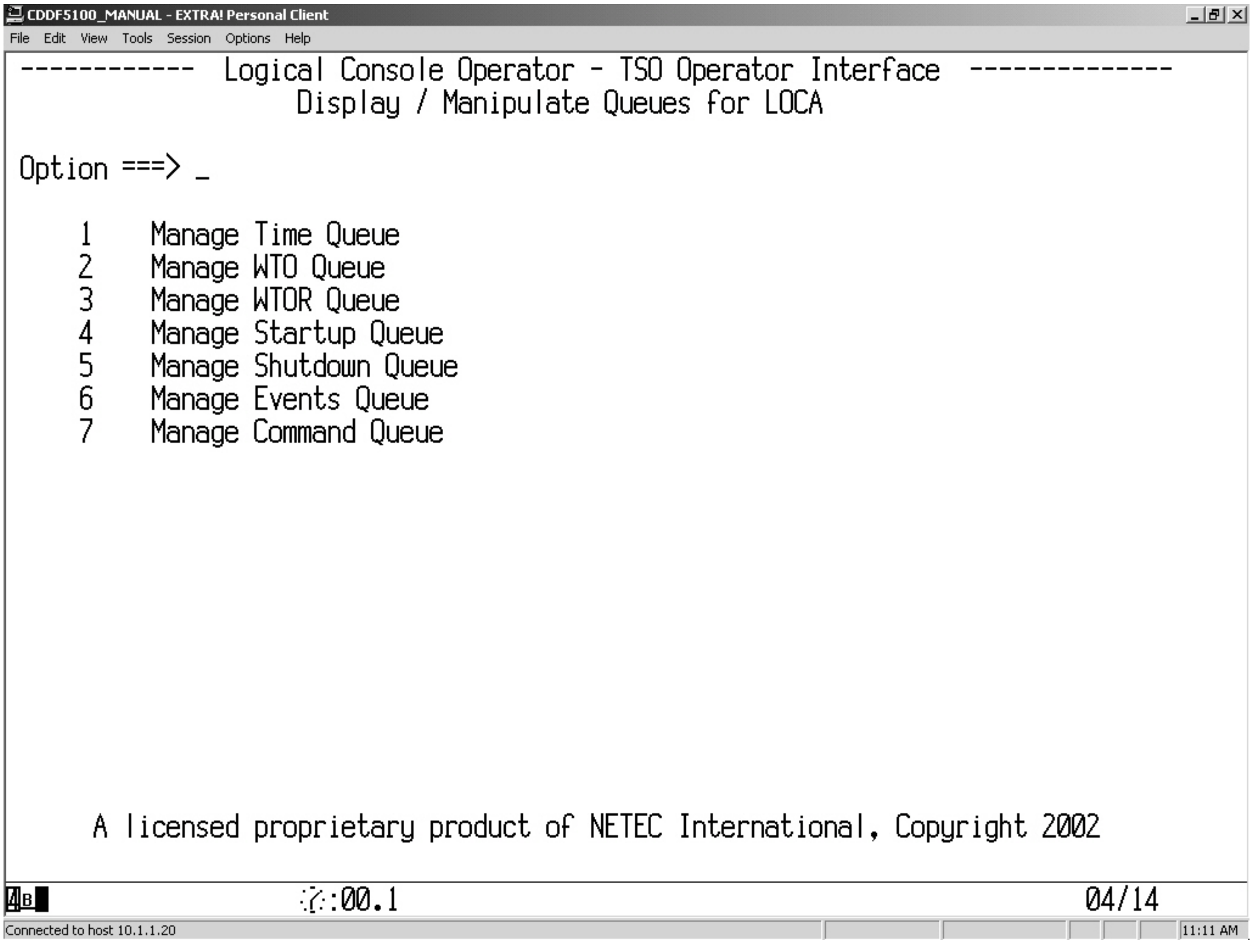
CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
LOCO SUBMIT DSNAME  NETEC.LOCO33.JCLLIB                        Row 00001 of 00019
Command ===> _                                                    Scroll ===> PAGE
   Name     Prompt          Size   Created         Changed             ID
. ASMLOCO                     97  2000/07/17  2002/08/10 12:03:08    U2TEC8
. ASM2016                     60  2001/11/15  2002/02/16 13:42:58    U2TEC8
. IEFBR14  *SUBMIT             0  2002/07/13  2002/08/07 18:17:48    U2TEC8
. LOCB                        25  2000/07/22  2002/08/25 10:47:23    U2TEC8
. MANYWTOS                     3  2002/04/13  2002/04/13 09:24:55    U2TEC8
. TASM                       197  2002/03/29  2002/03/29 13:42:18    U2TEC8
. TBLDR                       25  2000/07/22  2001/09/29 10:50:15    U2TEC8
. TCMD                         0  2001/09/03  2002/07/06 17:59:44    U2TEC8
. TDUMP                        6  2002/03/29  2002/03/29 12:54:03    U2TEC8
. TIGNT                       11  2002/03/29  2002/03/29 11:23:38    U2TEC8
. TINIT                       11  2000/07/22  2002/07/13 15:42:13    U2TEC8
. TINTF                       13  2001/09/01  2001/09/01 17:26:30    U2TEC8
. TLOG                        34  2001/09/03  2001/09/03 10:02:12    U2TEC8
. TREPT                       13  2001/07/03  2002/03/28 10:03:59    U2TEC8
. TSIMUL8                     23  2002/03/29  2002/03/29 16:24:09    U2TEC8
. TTEST                        6  2000/07/22  2001/09/01 08:20:15    U2TEC8
. TWQE                        23  2002/07/13  2002/07/13 14:55:11    U2TEC8
. TWTO                         5  2002/03/29  2002/03/29 14:16:03    U2TEC8
. ZAP                         12  2001/09/03  2001/09/03 11:20:23    U2TEC8
  **End**


4B                    :00.1                                              02/15
```

Connected to host 10.1.1.20 11:53 AM

Displaying and Manipulating LOCO Queues

The following screen is displayed when option “6” is entered from the selection menu. The screen lets the user select which LOCO queue he would like to manipulate.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 10.1.1.20
11:11 AM

```
------------  Logical Console Operator - TSO Operator Interface  --------------
                    Display / Manipulate Queues for LOCA

Option ===> _

    1    Manage Time Queue
    2    Manage WTO Queue
    3    Manage WTOR Queue
    4    Manage Startup Queue
    5    Manage Shutdown Queue
    6    Manage Events Queue
    7    Manage Command Queue





      A licensed proprietary product of NETEC International, Copyright 2002

4B                     :00.1                                           04/14
```

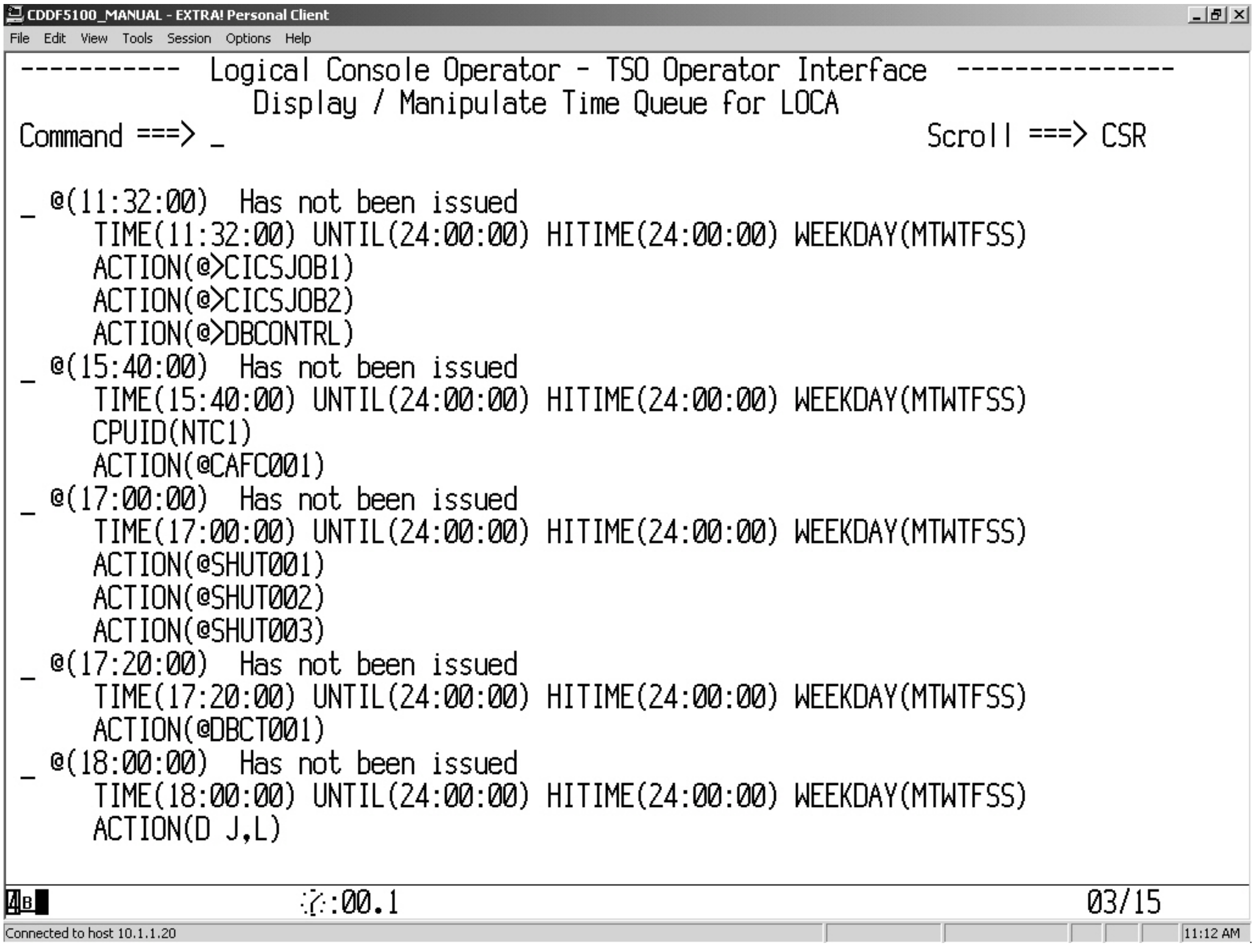
The queues that may be selected are:

Queue Name	Description
Time	Actions waiting for a specific time and date to occur.
WTO	Actions waiting for a specific WTO to be issued.
WTOR	Actions waiting for a specific WTOR to be issued.
Startup	Actions to occur at LOCO start up time.
Shutdown	Actions to occur a tLOCO shut down time.
Events	Actions to occur when specific end of job events occur.

Command Commands that may be issued by the user or as a target of a LOCO ACTION.

Displaying and Manipulating the TIME Queue

The following screen is displayed when the user selects the TIME queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
-----------  Logical Console Operator - TSO Operator Interface  ---------------
                Display / Manipulate Time Queue for LOCA
Command ===> _                                                  Scroll ===> CSR

_ @(11:32:00)  Has not been issued
     TIME(11:32:00) UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     ACTION(@>CICSJOB1)
     ACTION(@>CICSJOB2)
     ACTION(@>DBCONTRL)
_ @(15:40:00)  Has not been issued
     TIME(15:40:00) UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     CPUID(NTC1)
     ACTION(@CAFC001)
_ @(17:00:00)  Has not been issued
     TIME(17:00:00) UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     ACTION(@SHUT001)
     ACTION(@SHUT002)
     ACTION(@SHUT003)
_ @(17:20:00)  Has not been issued
     TIME(17:20:00) UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     ACTION(@DBCT001)
_ @(18:00:00)  Has not been issued
     TIME(18:00:00) UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     ACTION(D J,L)

4B                   :00.1                                      03/15
```

From this screen the user can delete, hold, or release a specific entry on the LOCO time queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following time queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the time queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the time queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this time queue entry. LOCO will remove the entry from the time queue. To replace the entry in the queue will require the LOCO table be reloaded.

Security Requirements

To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI - to access the LOCO subsystem.
LOCO.xxxx.QUEUES.TIME.BROWSE - to display queue entries.
LOCO.xxxx.QUEUES.TIME.HOLD - to HOLD/RELEASE queue entries.
LOCO.xxxx.QUEUES.TIME.DELETE - to DELETE queue entries.

xxxx = The one to four character LOCO subsytem name

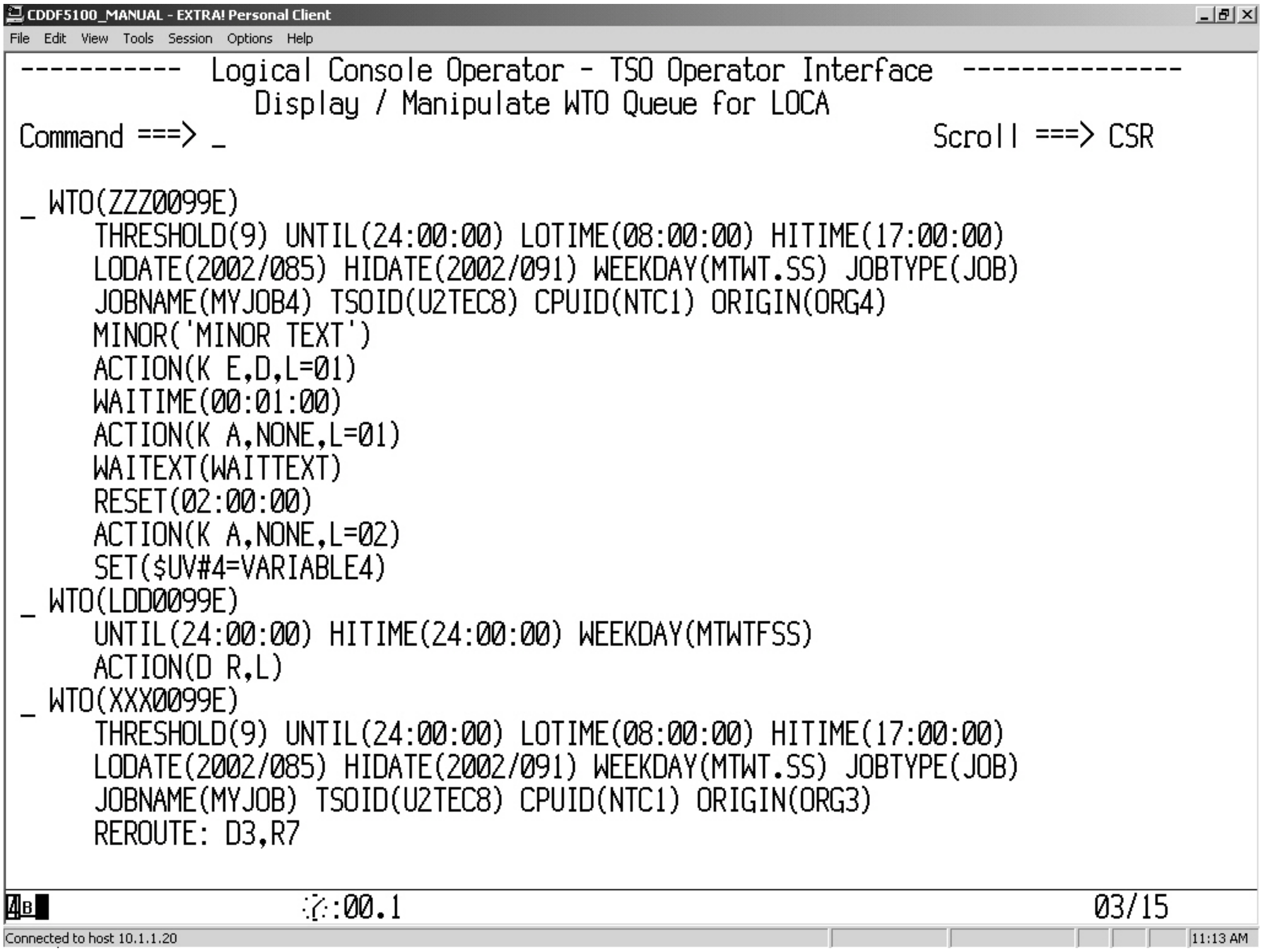
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI - to access the LOCO subsystem.
LOCO.xxxx.SUBMIT.BROWSE - to display the queue entries.
LOCO.xxxx.SUBMIT.EDIT - to edit the submit data set.
LOCO.xxxx.SUBMIT.SUBMIT - to request LOCO submit the job.

xxxx = The one to four character LOCO subsytem name

Displaying and Manipulating the WTO Queue

The following screen is displayed when the user selects the WTO queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
-----------  Logical Console Operator - TSO Operator Interface  ---------------
                Display / Manipulate WTO Queue for LOCA
Command ===> _                                                    Scroll ===> CSR

_ WTO(ZZZ0099E)
     THRESHOLD(9) UNTIL(24:00:00) LOTIME(08:00:00) HITIME(17:00:00)
     LODATE(2002/085) HIDATE(2002/091) WEEKDAY(MTWT.SS) JOBTYPE(JOB)
     JOBNAME(MYJOB4) TSOID(U2TEC8) CPUID(NTC1) ORIGIN(ORG4)
     MINOR('MINOR TEXT')
     ACTION(K E,D,L=01)
     WAITIME(00:01:00)
     ACTION(K A,NONE,L=01)
     WAITEXT(WAITTEXT)
     RESET(02:00:00)
     ACTION(K A,NONE,L=02)
     SET($UV#4=VARIABLE4)
_ WTO(LDD0099E)
     UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     ACTION(D R,L)
_ WTO(XXX0099E)
     THRESHOLD(9) UNTIL(24:00:00) LOTIME(08:00:00) HITIME(17:00:00)
     LODATE(2002/085) HIDATE(2002/091) WEEKDAY(MTWT.SS) JOBTYPE(JOB)
     JOBNAME(MYJOB) TSOID(U2TEC8) CPUID(NTC1) ORIGIN(ORG3)
     REROUTE: D3,R7

4B                    :00.1                                        03/15
Connected to host 10.1.1.20                                          11:13 AM
```

From this screen the user can delete, hold, or release a specific entry on the LOCO WTO queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following WTO queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the WTO queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the WTO queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this WTO queue entry. LOCO will remove the entry from the WTO queue. To replace the entry in the queue will require the LOCO table be reloaded.

Security Requirements

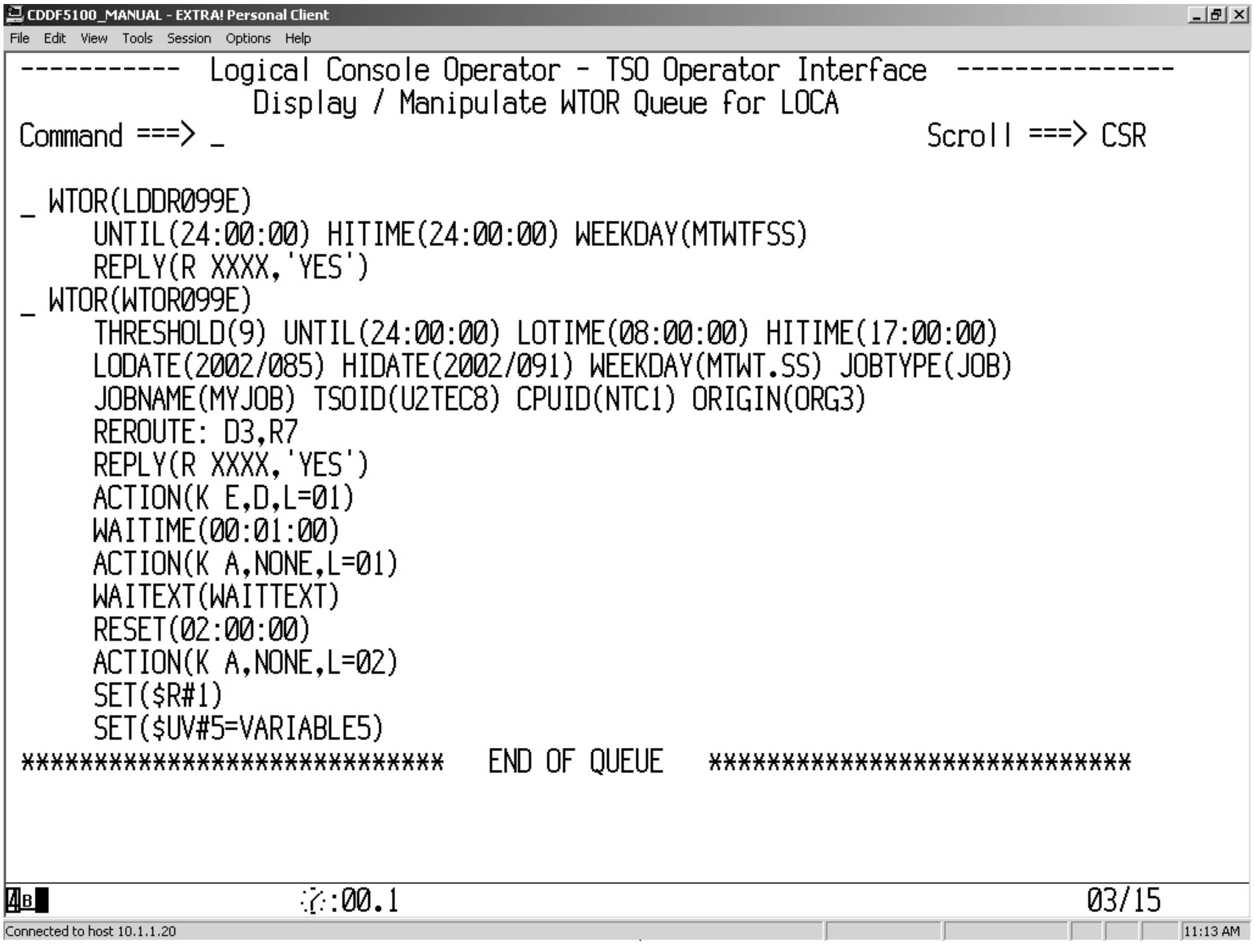
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI - to access the LOCO subsystem.
LOCO.xxxx.QUEUES.WTO.BROWSE - to display queue entries.
LOCO.xxxx.QUEUES.WTO.HOLD - to HOLD/RELEASE queue entries.
LOCO.xxxx.QUEUES.WTO.DELETE - to DELETE queue entries.

xxxx = The one to four character LOCO subsytem name

Displaying and Manipulating the WTOR Queue

The following screen is displayed when the user selects the WTOR queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
-----------  Logical Console Operator - TSO Operator Interface  ---------------
                Display / Manipulate WTOR Queue for LOCA
Command ===> _                                              Scroll ===> CSR

_ WTOR(LDDR099E)
     UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
     REPLY(R XXXX,'YES')
_ WTOR(WTOR099E)
     THRESHOLD(9) UNTIL(24:00:00) LOTIME(08:00:00) HITIME(17:00:00)
     LODATE(2002/085) HIDATE(2002/091) WEEKDAY(MTWT.SS) JOBTYPE(JOB)
     JOBNAME(MYJOB) TSOID(U2TEC8) CPUID(NTC1) ORIGIN(ORG3)
     REROUTE: D3,R7
     REPLY(R XXXX,'YES')
     ACTION(K E,D,L=01)
     WAITIME(00:01:00)
     ACTION(K A,NONE,L=01)
     WAITEXT(WAITTEXT)
     RESET(02:00:00)
     ACTION(K A,NONE,L=02)
     SET($R#1)
     SET($UV#5=VARIABLE5)
*****************************   END OF QUEUE   *****************************
```

From this screen the user can delete, hold, or release a specific entry on the LOCO WTOR queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following WTOR queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the WTOR queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the WTOR queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this WTOR queue entry. LOCO will remove the entry from the WTOR queue. To replace the entry in the queue will require the LOCO table be reloaded.

Security Requirements

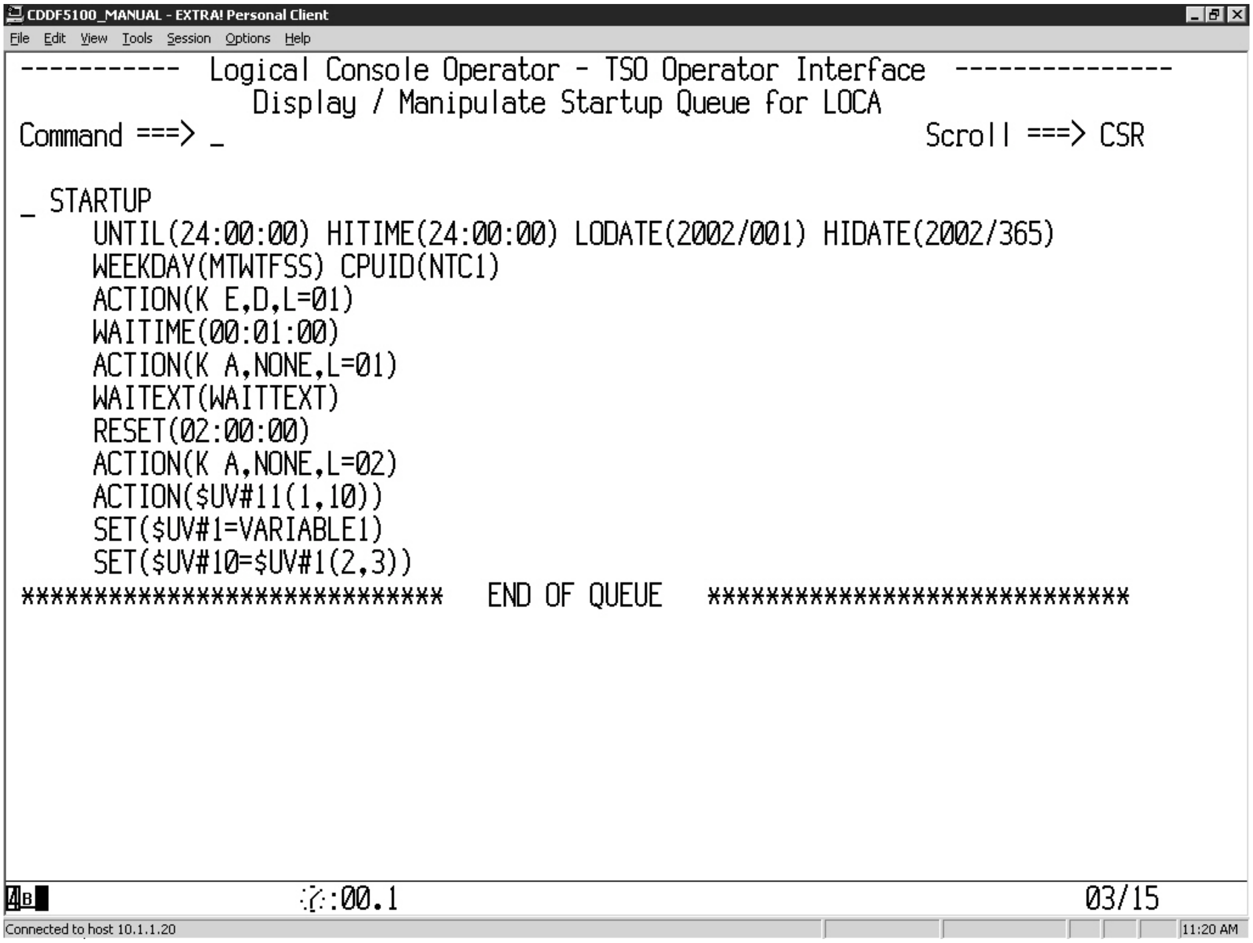
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI	-	to access the LOCO subsystem.
LOCO.xxxx.QUEUES.WTOR.BROWSE	-	to display queue entries.
LOCO.xxxx.QUEUES.WTOR.HOLD	-	to HOLD/RELEASE queue entries.
LOCO.xxxx.QUEUES.WTOR.DELETE	-	to DELETE queue entries.

xxxx = The one to four character LOCO subsytem name

Displaying and Manipulating the STARTUP Queue

The following screen is displayed when the user selects the STARTUP queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 10.1.1.20
11:20 AM

```
-----------  Logical Console Operator - TSO Operator Interface  ---------------
                Display / Manipulate Startup Queue for LOCA
Command ===> _                                                Scroll ===> CSR

_ STARTUP
     UNTIL(24:00:00) HITIME(24:00:00) LODATE(2002/001) HIDATE(2002/365)
     WEEKDAY(MTWTFSS) CPUID(NTC1)
     ACTION(K E,D,L=01)
     WAITIME(00:01:00)
     ACTION(K A,NONE,L=01)
     WAITEXT(WAITTEXT)
     RESET(02:00:00)
     ACTION(K A,NONE,L=02)
     ACTION($UV#11(1,10))
     SET($UV#1=VARIABLE1)
     SET($UV#10=$UV#1(2,3))
*****************************   END OF QUEUE   *****************************

4B                   :00.1                                             03/15
```

From this screen the user can delete, hold, or release a specific entry on the LOCO STARTUP queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following STARTUP queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the STARTUP queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the STARTUP queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this STARTUP queue entry. LOCO will remove the entry from the STARTUP queue. To replace the entry in the queue will require the LOCO table be reloaded.

Security Requirements

To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI - to access the LOCO subsystem.

LOCO.xxxx.QUEUES.STARTUP.BROWSE - to display queue entries.

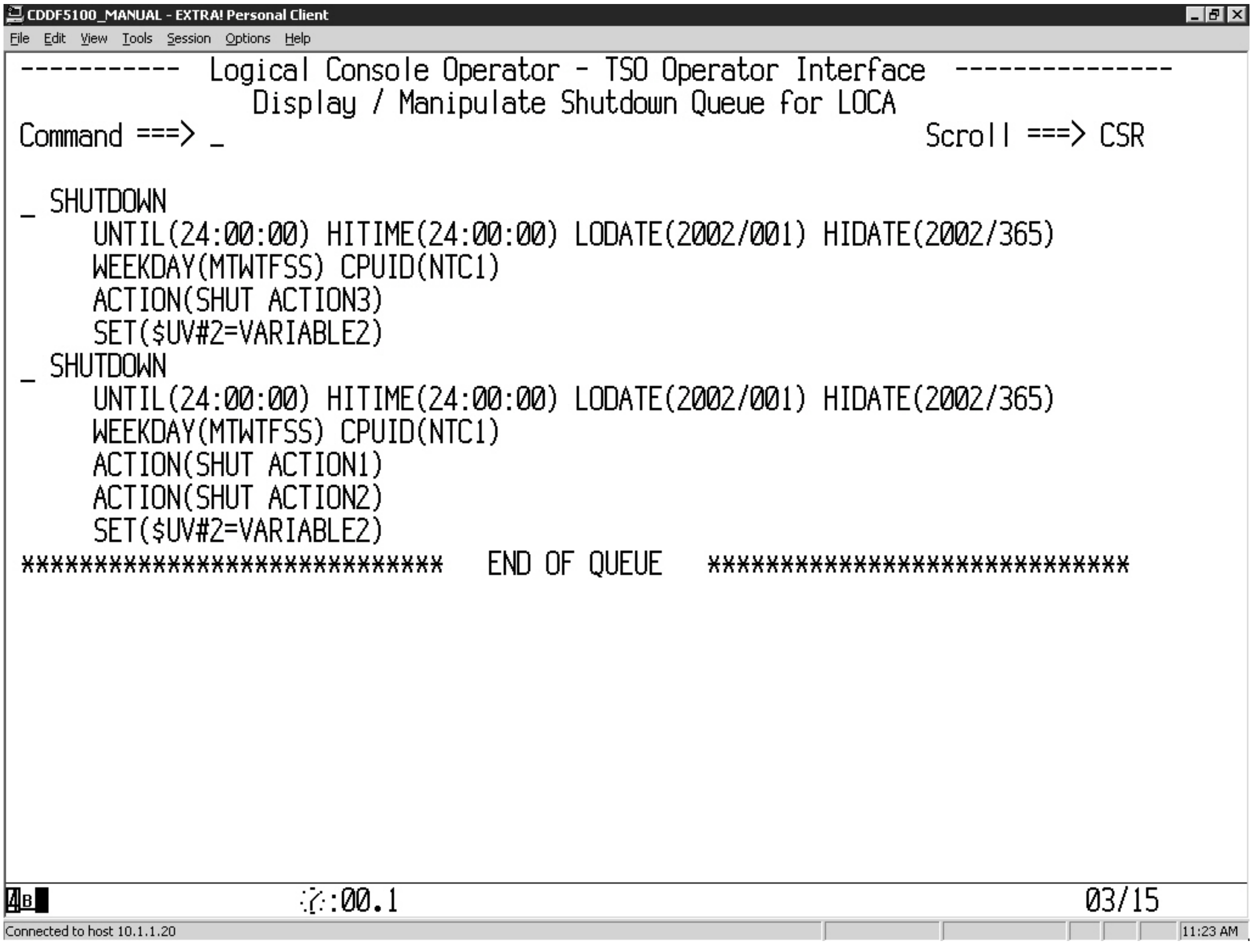
LOCO.xxxx.QUEUES.STARTUP.HOLD - to HOLD/RELEASE queue entries.

LOCO.xxxx.QUEUES.STARTUP.DELETE - to DELETE queue entries.

xxxx = The one to four character LOCO subsytem name

Displaying and Manipulating the SHUTDOWN Queue

The following screen is displayed when the user selects the SHUTDOWN queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
4B
:00.1
03/15
Connected to host 10.1.1.20
11:23 AM

```
-----------  Logical Console Operator - TSO Operator Interface  ---------------
                Display / Manipulate Shutdown Queue for LOCA
Command ===> _                                               Scroll ===> CSR

_ SHUTDOWN
     UNTIL(24:00:00) HITIME(24:00:00) LODATE(2002/001) HIDATE(2002/365)
     WEEKDAY(MTWTFSS) CPUID(NTC1)
     ACTION(SHUT ACTION3)
     SET($UV#2=VARIABLE2)
_ SHUTDOWN
     UNTIL(24:00:00) HITIME(24:00:00) LODATE(2002/001) HIDATE(2002/365)
     WEEKDAY(MTWTFSS) CPUID(NTC1)
     ACTION(SHUT ACTION1)
     ACTION(SHUT ACTION2)
     SET($UV#2=VARIABLE2)
*****************************   END OF QUEUE   *****************************
```

From this screen the user can delete, hold, or release a specific entry on the LOCO SHUTDOWN queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following SHUTDOWN queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the SHUTDOWN queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the SHUTDOWN queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this SHUTDOWN queue entry. LOCO will remove the entry from the SHUTDOWN queue. To replace the entry in the queue will require the LOCO table be reloaded.

Security Requirements

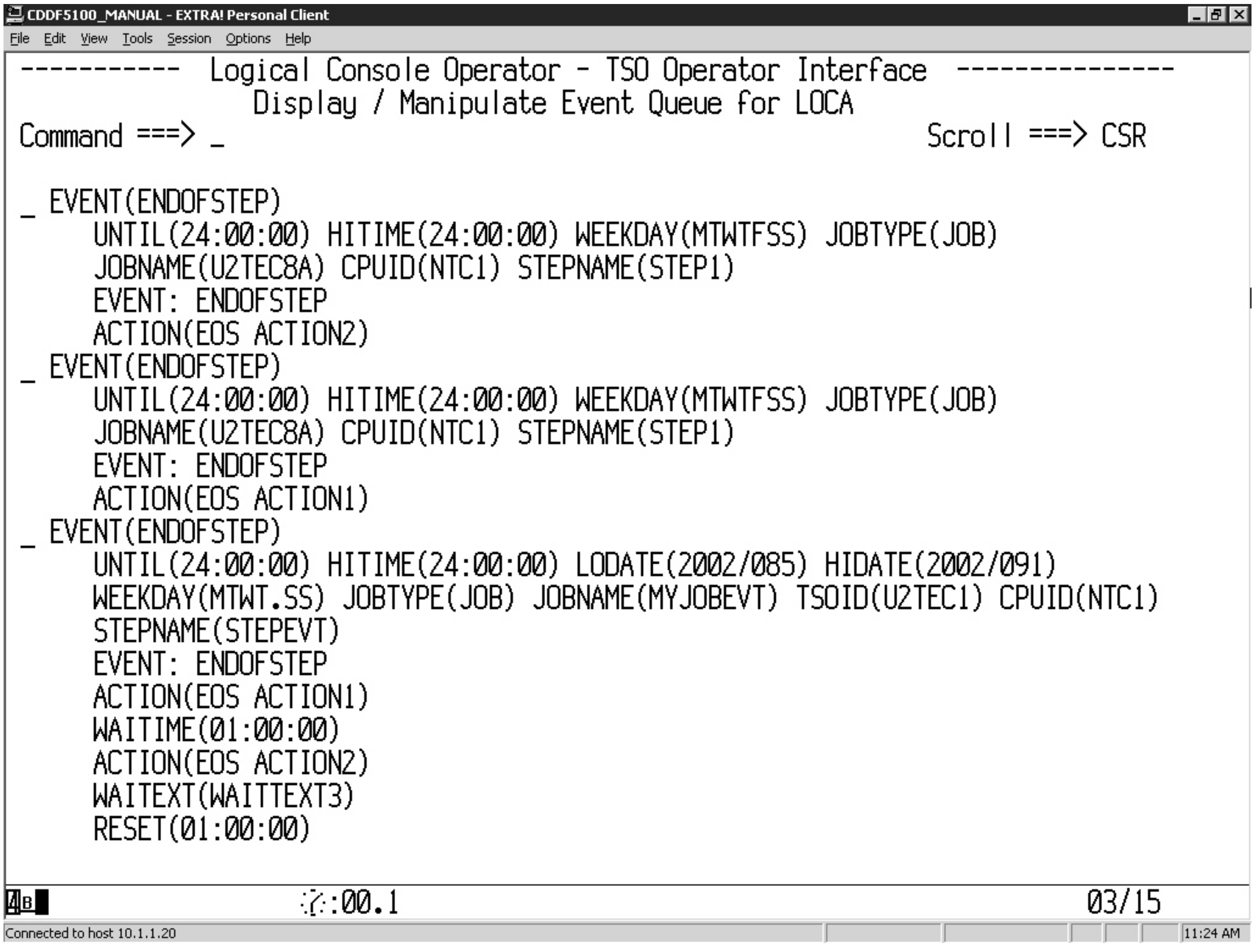
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI	- to access the LOCO subsystem.
LOCO.xxxx.QUEUES.SHUTDOWN.BROWSE	- to display queue entries.
LOCO.xxxx.QUEUES.SHUTDOWN.HOLD	- to HOLD/RELEASE queue entries.
LOCO.xxxx.QUEUES.SHUTDOWN.DELETE	- to DELETE queue entries.

xxxx = The one to four character LOCO subsytem name

Displaying and Manipulating the EVENT Queue

The following screen is displayed when the user selects the EVENT queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 10.1.1.20
11:24 AM

```
 -----------  Logical Console Operator - TSO Operator Interface  ---------------
                 Display / Manipulate Event Queue for LOCA
 Command ===> _                                                 Scroll ===> CSR

 _ EVENT(ENDOFSTEP)
      UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS) JOBTYPE(JOB)
      JOBNAME(U2TEC8A) CPUID(NTC1) STEPNAME(STEP1)
      EVENT: ENDOFSTEP
      ACTION(EOS ACTION2)
 _ EVENT(ENDOFSTEP)
      UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS) JOBTYPE(JOB)
      JOBNAME(U2TEC8A) CPUID(NTC1) STEPNAME(STEP1)
      EVENT: ENDOFSTEP
      ACTION(EOS ACTION1)
 _ EVENT(ENDOFSTEP)
      UNTIL(24:00:00) HITIME(24:00:00) LODATE(2002/085) HIDATE(2002/091)
      WEEKDAY(MTWT.SS) JOBTYPE(JOB) JOBNAME(MYJOBEVT) TSOID(U2TEC1) CPUID(NTC1)
      STEPNAME(STEPEVT)
      EVENT: ENDOFSTEP
      ACTION(EOS ACTION1)
      WAITIME(01:00:00)
      ACTION(EOS ACTION2)
      WAITEXT(WAITTEXT3)
      RESET(01:00:00)

4B                    :00.1                                          03/15
```

From this screen the user can delete, hold, or release a specific entry on the LOCO EVENT queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following EVENT queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the EVENT queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the EVENT queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this EVENT queue entry. LOCO will remove the entry from the EVENT queue. To replace the entry in the queue will require the LOCO table be reloaded.

Security Requirements

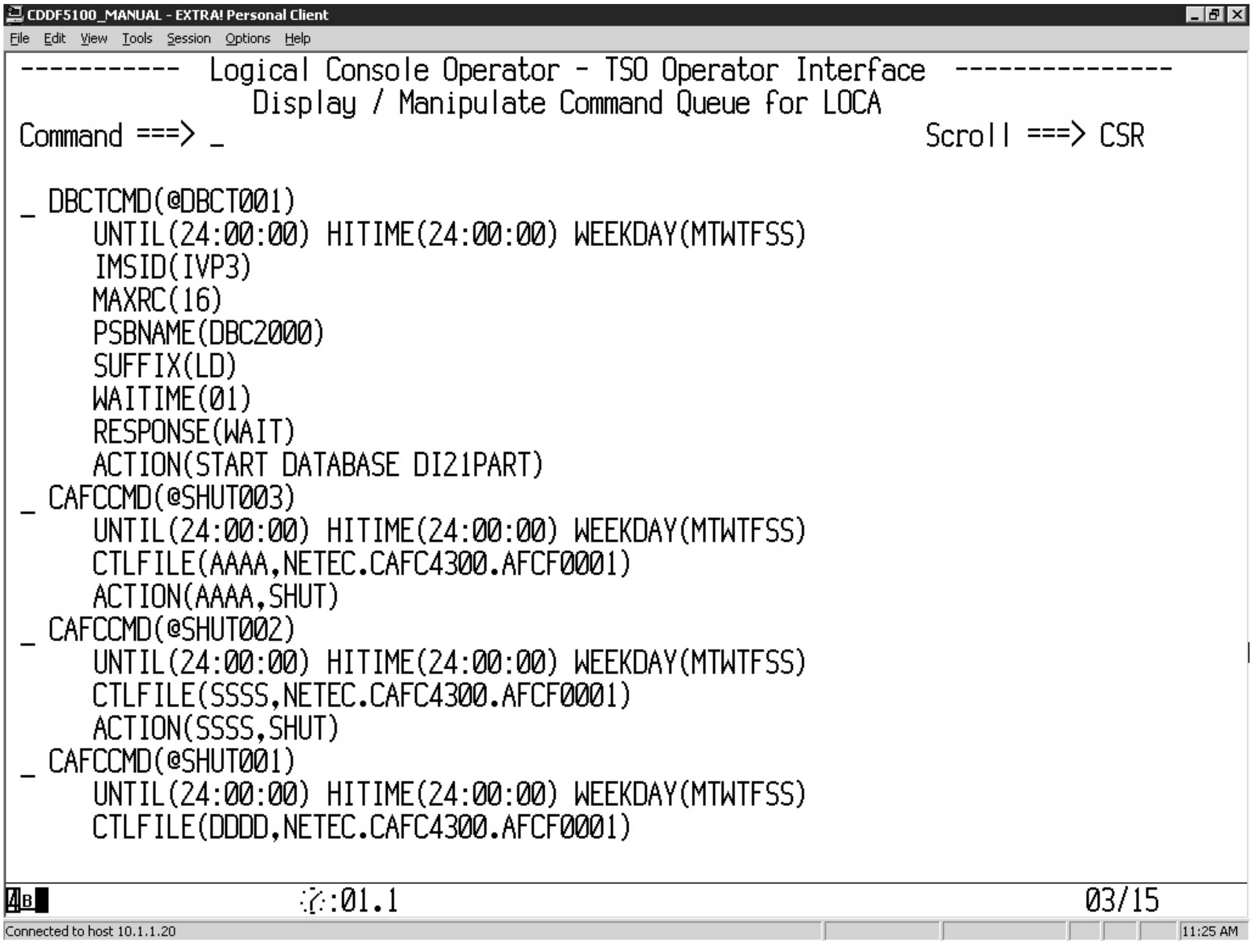
To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI	- to access the LOCO subsystem.
LOCO.xxxx.QUEUES.EVENT.BROWSE	- to display queue entries.
LOCO.xxxx.QUEUES.EVENT.HOLD	- to HOLD/RELEASE queue entries.
LOCO.xxxx.QUEUES.EVENT.DELETE	- to DELETE queue entries.

xxxx = The one to four character LOCO subsytem name

Displaying and Manipulating the COMMAND Queue

The following screen is displayed when the user selects the COMMAND queue for manipulation.


CDDF5100_MANUAL - EXTRA! Personal Client
File Edit View Tools Session Options Help

```
 -----------  Logical Console Operator - TSO Operator Interface  ---------------
                 Display / Manipulate Command Queue for LOCA
 Command ===> _                                                 Scroll ===> CSR

 _ DBCTCMD(@DBCT001)
      UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
      IMSID(IVP3)
      MAXRC(16)
      PSBNAME(DBC2000)
      SUFFIX(LD)
      WAITIME(01)
      RESPONSE(WAIT)
      ACTION(START DATABASE DI21PART)
 _ CAFCCMD(@SHUT003)
      UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
      CTLFILE(AAAA,NETEC.CAFC4300.AFCF0001)
      ACTION(AAAA,SHUT)
 _ CAFCCMD(@SHUT002)
      UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
      CTLFILE(SSSS,NETEC.CAFC4300.AFCF0001)
      ACTION(SSSS,SHUT)
 _ CAFCCMD(@SHUT001)
      UNTIL(24:00:00) HITIME(24:00:00) WEEKDAY(MTWTFSS)
      CTLFILE(DDDD,NETEC.CAFC4300.AFCF0001)

4B                    :01.1                                         03/15
```

From this screen the user can delete, hold, release, or execute a specific entry on the LOCO COMMAND queue. Each queue entry is formatted on multiple lines. Before the beginning of a queue entry is an action field where the user specifies what action he would like LOCO to take against the following COMMAND queue entry.

The valid actions that can be specified for a queue entry are:

Action	Description
H	Hold the COMMAND queue entry. Prevent LOCO from taking action should actions required to execute this queue entry be meet.

R Release the COMMAND queue entry. Release LOCO to take action should actions required to execute this queue entry be meet.

D Delete this COMMAND queue entry. LOCO will remove the entry from the COMMAND queue. To replace the entry in the queue will require the LOCO table be reloaded.

E Execute this COMMAND queue entry. LOCO will execute the command as it had been entered via the MVS console.

Security Requirements

To manipulate these resources requires READ access to the following SAF Profiles.

LOCO.xxxx.SSI	- to access the LOCO subsystem.
LOCO.xxxx.QUEUES.COMMAND.BROWSE	- to display queue entries.
LOCO.xxxx.QUEUES.COMMAND.HOLD	- to HOLD/RELEASE queue entries.
LOCO.xxxx.QUEUES.COMMAND.DELETE	- to DELETE queue entries.
LOCO.xxxx.QUEUES.COMMAND.EXECUTE	- to EXECUTE command queue.

xxxx = The one to four character LOCO subsytem name

Security for the LOCO TSO Interface

The LOCO TSO interface utilizes an external security manager such as RACF, ACF2, or Top Secret to verify a user’s functional authorization. SAF calls are made through out the interface to verify the user’s permission to execute the LOCO function that is being attempted. If the user is found to be unauthorized, access to the LOCO function is denied. Profile checking occurs at two levels. First the users ability to access the LOCO subsystem interface is determined. Second the users ability to execute the requested function within the specified LOCO subsystem is checked. If at any time, the user’s authorization fails, access to the requested function or LOCO subsystem is denied. Through the use of generic profiles, the LOCO security scheme can be as simple or as complex as the installation requires. The following paragraphs describe the SAF profiles that must be created to permit a user access to the LOCO TSO interface.

Security Profiles

The user must have read access to the SAF profile to be considered authorized for the requested function of subsystem. All profiles are defined in the FACILITY class.
The following describe the profiles that are checked for the function described.

Permission to access the LOCO subsystem interface

LOCO.ssss.SSI
ssss – Subsystem Name

Example

LOCO.*.SSI - Generic profile that will match any request to access any LOCO subsystem. Users permitted with READ access to this profile will be able to access any LOCO TSO interface, connecting to any LOCO subsystem.

LOCO.LOCA.SSI - Specific profile that will match users who are trying to access the LOCO subsystem named LOCA. Users permitted with READ access to this profile will be able to access the LOCO TSO interface, connecting to LOCO subsystem LOCA.

The RACF commands necessary to define these profiles are:

```
RDEFINE FACILITY LOCO.*.SSI      UACC(NONE)
RDEFINE FACILITy LOCO.LOCA.SSI   UACC(NONE)
```

Permission to access LOCO status screen

LOCO.ssss.STATUS
ssss – Subsystem Name

Example

```
LOCO.*.STATUS    - Generic profile that will match any
                   request to access the STATUS display for
                   any LOCO subsystem.  Users permitted
                   with READ access to this profile will be
                   able to access the status screen for any
                   LOCO TSO interface, connecting to any
                   LOCO subsystem.
LOCO.LOCA.STATUS - Specific profile that will match users
                   who are trying to access the status
                   screen for the LOCO subsystem named
                   LOCA.  Users permitted with READ access
                   to this profile will be able to access
                   status screen for LOCO subsystem LOCA.
```

The RACF commands necessary to define these profiles are:

```
RDEFINE FACILITY LOCO.*.STATUS     UACC(NONE)
RDEFINE FACILITy LOCO.LOCA.STATUS  UACC(NONE)
```

Permission to manipulate LOCO trace

LOCO.ssss.TRACES
ssss – Subsystem Name

Example

```
LOCO.*.TRACES    - Generic profile that will match any
                   request to modify the trace options for
                   any LOCO subsystem.  Users permitted
                   with READ access to this profile will be
                   able to modify the trace options for any
                   LOCO TSO interface, connecting to any
                   LOCO subsystem.
LOCO.LOCA.TRACES - Specific profile that will match users
                   who are trying to modify trace options
                   for the LOCO subsystem named LOCA.
```

```
                    Users permitted with READ access to this
                    profile will be able to modify trace
                    options for the LOCO TSO interface,
                    connecting to LOCO subsystem LOCA.
```

The RACF commands necessary to define these profiles are:

```
RDEFINE FACILITY LOCO.*.TRACES     UACC(NONE)
RDEFINE FACILITy LOCO.LOCA.TRACES  UACC(NONE)
```

Permission to manipulate LOCO resources

```
LOCO.ssss.RESOURCES.xxxxxxx
     ssss                  - Subsystem Name
               BROWSE      - User can examine the state of
                             LOCO resources
               HOLD        - User can HOLD and Release
                             resources
               ACTIVATE    - User can Activate and
                             deactivate resources
```

Example

```
LOCO.*.RESOURCES.BROWSE     -      Generic profile that will
                                   match any request to
                                   browse the LOCO resources
                                   on any LOCO subsystem.
                                   Users permitted with READ
                                   access to this profile
                                   will be able to examine
                                   the LOCO resources on any
                                   LOCO TSO interface,
                                   connecting to any LOCO
                                   subsystem.
LOCO.LOCA.RESOURCES.HOLD    -      Specific profile that
                                   will match users who are
                                   trying to access
                                   resources on the LOCO
                                   subsystem named LOCA.
                                   Users permitted with READ
                                   access to this profile
                                   will be able to hold and
                                   release resources on the
                                   LOCO TSO interface,
                                   connecting to LOCO
                                   subsystem LOCA.
```

The RACF commands necessary to define these profiles are:

```
     RDEFINE FACILITY LOCO.*.RESOURCES.BROWSE     UACC(NONE)
     RDEFINE FACILITy LOCO.LOCA.RESOURCES.HOLD    UACC(NONE)
```

Permission to submit jobs through LOCO

```
LOCO.ssss.SUBMIT.xxxxxx
     ssss                    - Subsystem Name
                 BROWSE      - User can browse the jobs
                               available for submission
                 EDIT        - User can edit jobs in the
                               LOCO submit library
                 SUBMIT      - User can request LOCO submit
                               jobs from the LOCO submit
                               library
```

Example

```
LOCO.*.SUBMIT.BROWSE         - Generic profile that will
                               match any request to access
                               any LOCO subsystem.  Users
                               permitted with READ access
                               to this profile will be able
                               to browse jobs in the LOCO
                               submit library of any LOCO
                               TSO interface, connecting to
                               any LOCO subsystem.
LOCO.LOCA.SUBMIT.SUBMIT      - Specific profile that will
                               match users who are trying
                               to access the LOCO subsystem
                               named LOCA.  Users permitted
                               with READ access to this
                               profile will be able to
                               submit jobs from the LOCO
                               submit library through the
                               LOCO TSO interface,
                               connecting to LOCO subsystem
                               LOCA.
```

The RACF commands necessary to define these profiles are:

```
     RDEFINE FACILITY LOCO.*.SUBMIT.BROWSE     UACC(NONE)
     RDEFINE FACILITy LOCO.LOCA.SUBMIT.SUBMIT  UACC(NONE)
```

Permission to manipulate LOCO tables

```
LOCO.ssss.TABLES.xxxxxx
          ssss               - Subsystem Name
```

```
                  BROWSE      - User can browse LOCO tables
                  EDIT        - User can edit LOCO tables
                  LOAD        - User can reload LOCO tables
                  IMPORT      - User can import table source
                                into the LOCO table library
                  EXPORT      - User can export table source
                                from the LOCO table library
```

Example

```
     LOCO.*.TABLES.BROWSE        - Generic profile that will
                                   match any request to access
                                   any LOCO subsystem.  Users
                                   permitted with READ access
                                   to this profile will be able
                                   to browse LOCO tables found
                                   in the LOCO table library of
                                   any LOCO TSO interface,
                                   connecting to any LOCO
                                   subsystem.
     LOCO.LOCA.TABLES.EDIT       - Specific profile that will
                                   match users who are trying
                                   to access the LOCO subsystem
                                   named LOCA.  Users permitted
                                   with READ access to this
                                   profile will be able to edit
                                   LOCO tables from the LOCO
                                   table library through the
                                   LOCO TSO interface,
                                   connecting to LOCO subsystem
                                   LOCA.
```

The RACF commands necessary to define these profiles are:

```
RDEFINE FACILITY LOCO.*.TABLES.BROWSE   UACC(NONE)
RDEFINE FACILITy LOCO.LOCA.TABLES.EDIT  UACC(NONE)
```

Permission to manipulate LOCO queues

```
     LOCO.ssss.QUEUES.qqqqqqq.xxxxxxxx
          ssss                        - Subsystem Name
                       qqqqqqq        - Queue Name
                                         WTO
                                         WTOR
                                         TIME
                                         STARTUP
                                         SHUTDOWN
                                         COMMAND
```

EVENT

xxxxxxxx – Action type

BROWSE - User can browse the LOCO queue.

HOLD - User can Hold and Release items in the LOCO queue.

DELETE - User can delete items from the LOCO Queue

EXECUTE - User can execute commands from the LOCO command queue display

Example

LOCO.*.QUEUES.WTO.BROWSE - Generic profile that will match any request to access any LOCO subsystem. Users permitted with READ access to this browse the WTO queuefor any LOCO TSO interface, connecting to any LOCO subsystem.

LOCO.LOCA.QUEUES.COMMANDS.DELETE - Specific profile that will match users who are trying to access the LOCO subsystem named LOCA. Users permitted with READ access to this profile will be able to delete commands from the LOCO command queue through the LOCO TSO interface, connecting to LOCO subsystem LOCA.

The RACF commands necessary to define these profiles are:

```
     RDEFINE FACILITY LOCO.*.QUEUES.WTO.BROWSE  UACC(NONE)
     RDEFINE FACILITy LOCO.LOCA.QUEUES.COMMANDS.DELETE
                                        UACC(NONE)
```

Hint

If no security is required for the LOCO TSO interface create a single generic profile of LOCO.* with a universal access of READ. This will allow all users access to all LOCO subsystems and access to all functions within the LOCO TSO interface.

Hints, Tips, Special Techniques and Examples

Communicating with CICS from LOCO

CICS is capable of accepting transactions from an MVS console via the MODIFY command. CICS can also accept MODIFY commands embedded in JCL streams or from ISPF under TSO. When the MODIFY commands come from a hardware console, they required the following TCTTE entry:

```
          DFHTCT  TYPE=TERMINAL,                 X
                  TRMIDNT=xxxx,                  X
                  TRMTYPE=CONSOLE,               X
                  CONSLID=nn
```

where xxxx is your choice of a four character terminal id and nn is the console-id of the particular console as display by the MVS console command: D CONSOLES.

When the MODIFY commands come from a non-hardware source like JCL, TSO or LOCO the required TCTTE entry is

```
          DFHTCT  TYPE=TERMINAL,                 X
                  TRMIDNT=xxxx,                  X
                  TRMTYPE=CONSOLE,               X
                  CONSLID=00
```

The CONSLID must be 00.

Before general transactions can be issued to the CICS console terminal, a sign-on must be issued for a valid user:

```
          F cicsjobname,CSSN NAME=name  PS=password
```

This sign-on must have a security level sufficient to issue any further CICS transactions you wish to start on the CICS console terminal.

CICS can be shutdown with

```
          F cicsjobname,CEMT S TAS ALL P
          F cicsjobname,CEMT P SHUT
```

The following LOCO table entries will accomplish the sign-on automatically and provide a short form command for shutdown:

```
          WTO(DFH1500 - ******** : CONTROL) -
          ACTION(F $JOBNAME,CSSN NAME=name PS=password)

          COMMAND(@SHUT********) -
               ACTION(F $TEXT(6,8),CEMT S TAS ALL P) -
               WAITIME(00:01:00) -
               ACTION(F $TEXT(6,8),CEMT P SHUT)
```

A shutdown at a certain time of day is similar:

```
          @(19:30:00) -
               ACTION(F cicsjobname,CEMT S TAS ALL P) -
               WAITIME(00:01:00) -
               ACTION(F cicsjobname,CEMT P SHUT)
```

Running LOCO before JES

LOCO may be started directly from COMMND00 member of SYS1.PARMLIB, and then used to start JES, reply to JES and start other IPL-time tasks. UNIT and VOL operands must be specified on the JCL DD statements. The following JCL PROC may be used to run a pre-JES LOCO:

```
     //STARTUP PROC P='MEMBER=LOCOIPL,SSN=LCOH'
     //IEFPROC   EXEC PGM=LOCOMAIN,
     //             PARM='&P',
     //             REGION=4M,
     //             TIME=1440,
     //             DPRTY=(15,15)
     //STEPLIB  DD  DISP=SHR,DSN=your.loco.loadlib,
     //             UNIT=nnnnn,VOL=SER=vvvvvv
     //TABLE    DD  DISP=SHR,DSN=your.loco.table.library,
     //             UNIT=nnnnn,VOLSER=vvvvvv
```

The COMMND00 member of SYS1.PARMLIB would contain COM='S STARTUP,SUB=MSTR' to invoke the LOCO proc. There will be an error message issued for missing DD cards for SYSPRINT and INTRDR; ignore these messages. This startup LOCO may not submit job streams since there is no INTRDR JCL to support job submission; but it can start tasks. The LOCO table member LOCOIPL could look something like:

```
STARTUP ACTION (S JES2) -
        ACTION (S RMF) -
        ACTION (S VTAMNET) -
        ACTION (S TSO) -
        ACTION (S UCC11) -
        WAITIME (00:00:30) -
        ACTION (S LOCO2) -
        ACTION (@STOP)
```

LOCO2 is the normal post-JES LOCO, which would have the full JCL for SUBMIT, INTRDR and SYSPRINT.

SLEEP verses WAITIME

The SLEEP operand an the WAITIME operand serve entirely different functions in LOCO. When all the requested ACTIONs of an LOCO table statement have been executed, LOCO recycles the statement to wait for its originating event (WTO or WTOR text). If SLEEP is coded on that statement, then LOCO deactivates the statement for the duration of the SLEEP interval before recycling it. Thus the table statement disappears for that period of time.

The WAITIME operand causes a time delay within a sequence of LOCO ACTIONs. Its purpose is to allow one ACTION to get started or perhaps to complete before starting the next ACTION.

It is important to understand that once an LOCO table statement is triggered, it must complete all the requested ACTIONs and be recycled before it can be triggered again. If that statement is SLEEPing or WAITingTIME when the next trigger event occurs, the trigger event will be lost.

Grouping related events

When events are interrelated by time interval, it is best to keep all the ACTIONs in a single LOCO table statement when possible. Suppose you want to start up a data collection task (like GTF trace) at a certain time, terminate it at a certain time, and then run some follow-up task to print the results. One way to do this is:

```
          @(16:00)  ACTION (S GTF.FRANK)
          WTOR (AHL125A)  REPLY (U)
          @(17:00)  ACTION (P FRANK) -
               WAITEXT (AHL006I) -
               ACTION (@>PRINTJOB)
```

In the event that the LOCO task itself starts execution between 16:00 and 17:00, the 17:00 event would be triggered. The P FRANK would have no effect since FRANK never got started; the WAITEXT for AHL006I would hang in a never-ending wait since AHL006I is produced by FRANK's termination which did not occur; and the

17:00 LOCO table statement would not be active for time triggering the next day as the clock passes through 16:00 and then through 17:00. A better set of LOCO statements would be:

```
          @(16:00)  ACTION (S GTF.FRANK)
                    WAITIME (01:00:00) -
                    ACTION (P FRANK) -
                    WAITEXT (AHL006I) -
                    ACTION (@>PRINTJOB)
          WTOR(AHL125A)  REPLY (END)
```

Here all the events controlling FRANK are contained in a single statement so that it is not possible for a segregated subset of ACTIONs to be triggered independently of the others.

Delayed reply to WTOR

It is possible to delay the REPLY to a WTOR in an LOCO table statement by inserting a WAITEXT before the REPLY:

```
          WTOR (IKT003D TCAS UNABLE) -
               WAITEXT (IST020I VTAM INITIALIZATION COMPLETE) -
               REPLY (RETRY)
```

Here TSO is trying to come up before VTAM is initialized. The WAITEXT delays the reply until VTAM initialization is complete.

Time Filters

LOTIME and HITIME operands may be coded singly or in pairs on LOCO statements. If LOTIME is coded and HITIME is not, then HITIME is considered to be the following midnight. If HITIME is coded and LOTIME is not, then LOTIME is considered to be the preceding midnight. If both are coded and the LOTIME value precedes the HITIME value, then a segment of the day is selected to consider triggering the associated event. If both are coded and HITIME precedes LOTIME, then two disjoint segments of the day are selected to consider triggering the associated event. An example of the last case is:

```
          WTO (ABC123) ACTION (S TASK047) -
               LOTIME (20:00:00) HITIME (04:00:00)
```

Here the desired time range is 8 PM until 4 AM each night. So in any given 24 hour day (midnight to midnight), there is an early morning period of midnight until 4 AM and a late night period of 8 PM until midnight. A WTO of ABC123 occurring during either of these two periods will trigger the ACTION.

Date Filters

LODATE and HIDATE operands may be coded singly or in pairs on LOCO statements. If LODATE is coded without HIDATE, then the range of dates consists of the LODATE value and all future dates. If HIDATE is coded without LODATE, then the range of dates consists of the HIDATE value and all previous dates. In all cases the end points are inclusive.

Now suppose you have a situation such that you want an LOCO defined event to be triggered only from 10 AM on Friday October 24 until 4 PM on Monday October 27. You might be tempted to code the LOCO statement:

```
WTO (ABC123) ACTION (S TASK23X) -
     LODATE (10/24/1999) LOTIME (10:00:00) -
     HIDATE (10/27/1999) HITIME (16:00:00)
```

This statement would appear to span the desired time range. However, the time range of 10:00 to 16:00 will be check for each individual day within the date range of 10/24 to 10/27. With a little ingenuity the original intent can be obtained:

```
WTO (ABC123) ACTION (S TASK23X) -
     DATE (10/24/1999) LOTIME (10:00:00)
WTO (ABC123) ACTION (S TASK23X) -
     LODATE (10/25/1999) -
     HIDATE (10/26/1999)
WTO (ABC123) ACTION (S TASK23X) -
     DATE (10/27/1999) HITIME (16:00:00)
```

Here the beginning and ending days are separated as specific DATEs with a beginning and ending time respectively, and the middle days have no time periods.

NOT CATLGD 2 messages

The system message IEF287I is used to indicate that a data set was not catalogued during deallocation processing at step termination. It is not normally a WTO, but rather a WTP (write to programmer) and appears only in the data set disposition message log. If you wish to issue LOCO ACTIONs in the event of NOT CATLGD 2 condition, you may cause the IEF287I message to be routed to the console as a WTO by issuing the MVS command MN STATUS. The sample LOCO table member IEF287I shows how to code

LOCO ACTIONs for message IEF287I. In the event that you do not want the IEF287I message to actually appear on your console, you need only ask LOCO to suppress it by adding the operand SUPPRESS to the table entry for WTO (IEF287I....).

An alternate method of causing IEF287I messages to be routed to the console is to ZAP the routing bits in IEFAB4M6, which is a message table residing in SYS1.LPALIB:

```
     DUMP IEFW21SD IEFAB4M6
     NAME IEFW21SD IEFAB4M6
     VER 00BE 20D5,D6E340C3,C1E3D3C7,C44040E7
     VER 00CC 10000020
     REP 00CC 10004020
```

Deferred reply to WTORS

It is possible to defer the reply to a WTOR by maintaining the reply number in an LOCO reply $var. Suppose that in an LOCO 'shutdown' procedure you wish to terminate NCCF. NCCF requires a reply of 'CLOSE IMMED', but the reply cannot be specified directly because you do not know the reply number in advance. LOCO can obtain and maintain the latest reply number for NCCF. The LOCO control statement below saves the most recent outstanding reply number for NCCF.

```
     WTOR(DSI802A NCCF   REPLY) SET($R#9)
```

Since it is possible that NCCF could already be running and have already issued its WTOR at the time LOCO is started, its prudent to ask LOCO to automatically 'prime' the reply $var for currently existing WTORs at each and every LOCO's startup. The following LOCO control statement will accomplish this:

```
     PRIME(DSI802A NCCF  REPLY) SET($R#9)
```

The PRIME statement is activated (1) at every LOCO startup, and (2) whenever a new LOCO control table is @LOADED.

The following ACTION, as a part of your 'shutdown' ACTIONs will gracefully terminate NCCF:

```
     ACTION(R $R#9,CLOSE IMMED)
```

At the time the above ACTION is issued, the reply $var $R#9 will be replaced with the current four digit reply number.

Utilizing LOCO to initiate CAFC or DBCTL events

LOCO can be used to initiate CICS Application File Control (CAFC) or DB Control (DBCTL)events either based on time of day or the occurrence of external events such as recognition of specific WTOs. This function of LOCO requires that the user be licensed for the NETEC CAFC product for CAFC functions and the NETEC DBCTL product for DB Control functions. The following is an example of a LOCO table that will cause CAFC or DBCTL events.

Example

In this example the user wants to automatically ask CAFC to close the CICS files QAFCT01 and QAFCT02 in the CICS99P region and close the CICS file QAFCT03 in the CICS99T region at 01:00 pm every day. The CAFC control file dsname for CICS99P is NETEC.VSAM.CAFC4300.AFCF001 and the dsname for the CICS99T control file is NETEC.VSAM.CAFC4300.AFCF001. The user also wants to start the DI21PART database when the IMS message DFS995I WARM START COMPLETE is issued by the DB control with job name DBCTL.

```
*------------------------------------------------------------* 00010000
*                                                            * 00020000
*                  PRODUCTION LOCO TABLE                     * 00030000
*                                                            * 00040000
*------------------------------------------------------------* 00050000
DEFINE  CARDSIZE(72) -                                         00060000
    PREFIX(@) -                                                00070000
    QLIMIT(200) -                                              00080000
    SLEEP(00:00:00) -                                          00090000
    WILDCARD(*)                                                00100000
*------------------------------------------------------------* 00180001
*                                                            * 00190001
*               CAFC COMMAND CONSTRUCTION                    * 00200001
*                                                            * 00210001
*------------------------------------------------------------* 00220001
CAFCCMD(@CAFC001)                                     -        00230001
       CTLFILE(CICS99P,NETEC.VSAM.CAFC4300.AFCF0001) -        00250010
       CTLFILE(CICS99T,NETEC.VSAM.CAFC4300.AFCF0001) -        00251010
       ACTION(CICS99P,C,DD,QAFCT01)                   -        00260010
       ACTION(CICS99P,C,DD,QAFCT02)                   -        00270010
       ACTION(CICS99T,C,DD,QAFCT03)                            00280010
*------------------------------------------------------------* 00180001
*                                                            * 00190001
*              DB CONTROL COMMAND CONSTRUCTION               * 00200001
*                                                            * 00210001
*------------------------------------------------------------* 00220001
DBCTCMD(@DBCT001)                                     -        00530001
       IMSID(IVP3)                                    -        00540001
       MAXRC(16)                                      -        00541008
       PSBNAME(DBC9000)                               -        00542008
       SUFFIX(LD)                                    -        00543008
       WAITIME(1)                                    -        00544008
       RESPONSE(WAIT)                                -        00545009
       ACTION(START DATABASE DI21PART)                        00550001
*-----------------------------------------------------------* 00180001
*                                                           * 00190001
*                     TIMED ACTIONS                         * 00200001
*                                                           * 00210001
*-----------------------------------------------------------* 00220001
@(13:00:00)                                          -        00570002
       ACTION(@CAFC001)                                       00570003
*-----------------------------------------------------------* 00180001
*                                                           * 00190001
*                 WTO INITIATED ACTIONS                     * 00200001
*                                                           * 00210001
*-----------------------------------------------------------* 00220001
WTO(DFS994I WARM START COMPLETED.)                   -        00100900
    JOBNAME(DBCTL)                                   -        00101000
    ACTION(@DBCT001)                                          00101100
```

Error Messages

Messages from LOCOBLDR (LCON0001-LCON0099)

NOT ABLE TO OPEN xxxxxxxx

Explanation: Probably missing DD card for ddname xxxxxxxx.

User Response: Supply missing DD card.

LCON0001 EXPECTED STATEMENT KEYWORD...FOUND xxxxxxxxxxxx

Explanation: xxxxxxxxxxxx is not a valid LCON statement keyword. Valid statement keywords are: WTO, WTOR, COMMAND, DEFINE, STARTUP, @.

User Response: Correct the invalid statement keyword. See LCON USER GUIDE for valid keyword usage. If you did not intend for a new LCON statement to begin with the specified keyword, then there may be a missing continuation indicator on the previous line.

LCON0002 xxxxxxxxxxxx IS INCORRECT STRUCTURE FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy is not in the required structural format for the keyword xxxxxxxxxxxx. If the keyword value contains a clock value or time of day, then too many colons (:) were found.

User Response: Correct the structural error.

LCON0003 EXPECTED OPERAND KEYWORD ...FOUND xxxxxxxxxxxx

Explanation: The statement syntax checker expected to find an operand keyword, but found the text xxxxxxxxxxxx instead. There are many operand keywords. Some of them are: WILDCARD, CARDSIZE, PREFIX, JOBNAME, DATE, LODATE, HIDATE, WEEKDAY, LOTIME, HITIME, REPEAT, REPLY, SLEEP, THRESHOLD, ACTION, RESET, WAITIME, WAITEXT, ROUTE, REROUTE, SUPPRESS.

User Response: Correct the invalid keyword. This is most likely a spelling error or a missing continuation indicator on the previous line.

LCON0004 LCON BUILDER VERSION x.x (mm/dd/yyyy)

Explanation: x.x is the LCON release number. mm/dd/yyyy is the assembly date of the program module LOCOBLDR.

User Response: None.

LCON0005 OPERAND VALUE IS NULL
Explanation: No value was specified for the operand.
User Response: Provide the proper value for the operand.

LCON0006 SYSIN FILE IS EMPTY
Explanation: An end of file was found on SYSIN on the first read.
User Response: Change the SYSIN DD JCL to reference an LCON source control table.

LCON0007 CARD IS EMPTY
Explanation: A blank card was encountered.
User Response: Change the blank card into a comment by placing an asterisk (*) in column 1.

LCON0008 KEYWORD IS NULL
Explanation: The statement syntax checker expected to find an operand keyword, but found a left parentheses instead.
User Response: Place the appropriate keyword before the left parentheses.

LCON0009 KEYWORD IS TOO LONG...FOUND xxxxxxxxxxxx
Explanation: The statement syntax checker expected to find an operand keyword, but found a string of text which is too long to be a keyword operand.
User Response: Correct the incorrect keyword operand.

LCON0010 EXPECTED CONTINUATION CARD...DISAPPOINTED BY EOF
Explanation: The previous statement indicated a continuation, but end of file was encountered instead.
User Response: Either provide the proper continuation card, or remove the continuation indicator from the previous card.

LCON0011 EXPECTED '('...BEFUDDLED BY GARBAGE xxxxxxxxxxxx
Explanation: The statement syntax checker expected to find a left parentheses, but found xxxxxxxxxxxx instead. There must always be a '(' between a keyword and its value.
User Response: Provide the left parentheses.

LCON0012 xxxxxxxxxxxx IS LESS THAN MIN VALUE...FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy for keyword operand xxxxxxxxxxxx is less than the minimum acceptable value.

User Response: Correct the operand value.

LCON0013 xxxxxxxxxxxx VALUE SPANS TOO MANY CARDS

Explanation: The value of an operand may span at most two cards.

User Response: Rearrange the operand keyword and value so that no more than two cards are spanned.

LCON0014 xxxxxxxxxxxx EXCEEDS MAX VALUE FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy for keyword operand xxxxxxxxxxxx is greater than the maximum acceptable value.

User Response: Correct the operand value.

LCON0015 xxxxxxxxxxxx MUST BE NUMERIC FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy for keyword operand xxxxxxxxxxxx is required to be numeric, but it is not.

User Response: Correct the operand value.

LCON0016 xxxxxxxxxxxx VALUE IS UNKNOWN FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy for keyword operand xxxxxxxxxxxx is not an acceptable value.

User Response: Correct the operand value.

LCON0017 xxxxxxxxxxxx VALUE IS INCORRECT LENGTH FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy for keyword operand xxxxxxxxxxxx contains more or less than the required number of bytes.

User Response: Correct the operand value.

LCON0018 xxxxxxxxxxxx MUST BE MM/DD/YY OR MM/DD/YYYY ...FOUND yyyyyyyyyyyy

Explanation: The value yyyyyyyyyyyy for keyword operand xxxxxxxxxxxx must be in the format mm/dd/yy or mm/dd/yyyy exactly, but it is not.

User Response: Correct the operand value to exactly 8 bytes with the character '/' in positions 3 and 6. Positions 1 - 2 must be numeric in the range of 01 thru 12. Positions 4 - 5 must be numeric in the range of 01 thru 28, or 01 thru 31 as appropriate for the month. Century portion of YYYY must be 19 or 20.

LCON0019 xxxxxxxxxxxx NOT ALLOWED IN THIS STATEMENT

Explanation: The keyword operand xxxxxxxxxxxx may not be used in this LCON statement.

User Response: See the LCON USERS GUIDE for the proper keyword combinations for this statement.

LCON0020 xxxxxxxxxxxx REQUIRED IN THIS STATEMENT

Explanation: The keyword operand xxxxxxxxxxxx must be used in this LCON statement.

User Response: See the LCON USERS GUIDE for the proper keyword combinations for this statement.

LCON0021 xxxxx ERRORS DETECTED IN STATEMENTS

Explanation: The syntax checker found xxxxx total errors in the LCON statements.

User Response: If errors were found, correct them and recompile the LCON table with LOCOBLDR.

LCON0022 SORT ERROR

Explanation: LOCOBLDR links the your standard sort program. There should be error messages from the sort pertaining to this error. Sort messages go to the SYSOUT DD file.

User Response: Check your sort messages manual for corrective action.

LCON0023 xxxxxxxxxxxx and yyyyyyyyyyyy ARE MUTUALLY EXCLUSIVE OPERANDS

Explanation: The keyword operand xxxxxxxxxxxx and yyyyyyyyyyyy may not both be used in an LCON statement.

User Response: See the LCON USERS GUIDE for the proper keyword combinations for this statement.

LCON0025 xxxxxxxxxxxx SPECIFIED TWICE WITHOUT INTERVENING ACTION
Explanation: WAITIME or WAITEXT may be used only once each between LCON ACTIONs.
User Response: See the LCON USERS GUIDE for the proper keywordcombinations for this statement.

LCON0026 xxxxxxxxxxxx CONTAINS INCORRECT ROUTE/DESCRIPTOR CODES
Explanation: Route codes may be any value from this list: R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16 Descriptor codes may be any value from this list: D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16 The codes may be combined in any sequence separated by commas. E. G. (R4,D2,R8,D1)
User Response: See the LCON USERS GUIDE for the proper specification of route and descriptor codes.

LCON0027 UNABLE TO FIND TERMINATING) FOR ROUTE/DESCRIPTOR CODES
Explanation: The syntax checker failed to find the end of the list of route/descriptor codes. The list should be terminated with a right parentheses.
User Response: Correct the route/descriptor list by placing a right parentheses at the end of it.

LCON0028 RESET MUST BE PRECEDED BY WAITEXT
Explanation: The RESET operand keyword is only used in conjunction with the WAITEXT operand keyword. RESET is to be placed after WAITEXT.
User Response: Correct the RESET/WAITEXT operand keywords.

LCON0029 RESET AND WAITIME ARE MUTUALLY EXCLUSIVE OPERANDS IN THIS CONTEXT
Explanation: RESET and WAITIME may not be both used to terminate a WAITEXT event.
User Response: Use either RESET or WAITIME operand keywords, but not both.

LCON0030 xxxxxxxx SUBSTRING CANNOT BEGIN A ZERO
Explanation: A $variable symbol substring expression must begin at position 1 or greater.
User Response: Correct the substring expression beginning position.

LCON0031 EXPECTED SOME VALUE ON RIGHT SIDE OF = SYMBOL
Explanation: In a SET operand for a user $variable the string expression on the right for the = symbol was null.
User Response: Provide a non-null string expression for the SET operand.

LCON0032 XXXXXXXX SUBSTRING EXPRESSION HAS INVALID SYNTAX
Explanation: The required syntax for $variable substring expression is: xxxxxx(n,m) where xxxxxx is the $variable label, n is the beginning position and m is the length
User Response: Provide a correct substring expression the $variable.

LCON0033 XXXXXXXX SUBSTRING EXCEEDS MAXIMUM LENGTH
Explanation: The requested substring extraction represents characters beyond the maximum length of the particular $variable. If L is the maximum length of $variable xxxxxx, then your substring xxxxxx(n,m) is such that n+m-1>L.
User Response: Provide a correct substring expression the $variable.

LCON0034 XXXXXXXX NOT ALLOWED IN THIS STATEMENT TYPE
Explanation: The $variable xxxxxxxx is not allowed in this statement because it cannot be evaluated. Local $variables require some triggering text (WTO, WTOR or COMMAND).
User Response: Remove the offending $variable.

LCON0035 SYNTAX ERROR IN TSOID OF SEND COMMAND
Explanation: The required format to specify a TSOID in an LCON SEND command is: @$(ttttttt)xxxxxxxxxx. The @ is your defined LCON command prefix; the $ is the send command; the () surround the TSOID of the intended recipient ttttttt; xxxxxxxxxx is the message text.
User response: Specify a correct syntax.

LCON0036 SEND COMMAND HAS NO MESSAGE TEXT
Explanation: The text of the SEND command was found to be null.
User response: Provide some text to make the message more meaningful to the recipient.

LCON0037 EXPECTED USER $VARIABLE AS TARGET

Explanation: In a SET operand the first characters must represent the label of the target $variable to receive the desired string.

User Response: Provide a target $variable in the SET operand.

LCON0038 EXPECTED = SYMBOL IMMEDIATELY AFTER TARGET $VARIABLE

Explanation: In a SET operand the target $variable label must be followed by an = symbol. Example: SET (#UV#23=xxxyyy)

User response: Provide an = symbol; it is generally in the upper right portion of the keyboard.

LCON0039 REPLY $VARIABLE MUST STAND ALONE

Explanation: In a SET operand for a reply $variable extraneous data was found. The required syntax is SET ($R#nn).

User response: Provide a correct syntax for the SET operand.

LCON0040 ACTION TOO LONG; MAX LENGTH IS 120

Explanation: ACTION text was found to be greater than 120 bytes.

User Response: Reduce the length of the ACTION text.

LCON0041 EVENT NAME UNKNOWN

Explanation: An EVENT statement specified an event unknown to LCON. Currently ENDOFSTEP is the only valid event

User Response: Correct the EVENT statement.

LCON0042 EXPECTED (SYMBOL PRECEEDING CALLED PROGRAM NAME

Explanation: LOCOBLDR found exit program call in an ACTION, but the syntax was incorrect.

User Response: Correct the syntax.

LCON0043 UNABLE TO FIND TERMINATING) SYMBOL AFTER CALLED PROGRAM NAME

Explanation: LOCOBLDR found exit program call in an ACTION, but the syntax was incorrect.

User Response: Correct the syntax.

LCON0044 CALLED PROGRAM NAME IS NULL

Explanation: LOCOBLDR found exit program call in an ACTION, but the syntax was incorrect. The exit program name was missing.

User Response: Correct the syntax.

LCON0045 USER HAS SPECIFIED A RESERVED COMMAND NAME

Explanation: LOCOBLDR found that a LOCO reserved command name was specified for a user defined command.

User Response: Change the user specified command name.

LCON0046 USER COMMANDS CAN NOT START WITH SPECIAL CHARACTERS

Explanation: LOCOBLDR found that a user defined LOCO command name began with a special character.

User Response: Remove the spcial character from the beginning of the user defined command.

LCON0047 USER HAS SPECIFIED A DUPLICATE COMMAND NAME

Explanation: LOCOBLDR found that a LOCO command name was specified more than once.

User Response: Create a new command name for one of the user defined commands.

LCON0048 GETMAIN FAILURE

Explanation: LOCOBLDR has encountered an internal error while processing.

User Response: Rerun LOCOBLDR with a larger region size. If the problem continues contact NETEC Technical Support.

Messages from LOCOMAIN (LCON0101-LCON0300)

Messages are shown with LCON as the message prefix. This prefix will be changed to the current subsystem name by LOCOMAIN at execution time. If LOCOMAIN is run with PARM SSN=LCON, then the messages will be LCON101, LCON102 etc.

NOT ABLE TO OPEN xxxxxxxx
Explanation: Probably missing DD card for ddname xxxxxxxx.
User Response: Supply missing DD card.

PARM VALUE TOO LARGE, MAX SIZE = 79 BYTES
Explanation: Parm value is too large. It may not be longer than 79 bytes.
User Response: Remove extraneous data from PARM.

LCON0101 SHUTDOWN AT CONSOLE REQUEST
Explanation: LCON has been requested to shutdown by the @STOP command.
User Response: None.

LCON0102 NOT ABLE TO OPEN TABLE FILE
Explanation: Probably missing DD card for ddname TABLE.
User Response: Supply missing DD card.

LCON0103 NOT A LCON TABLE
Explanation: The member specified in the PARM to LOCOMAIN or in the @LOAD command is not an LCON object table.
User Response: The program module LOCOMAIN will only process LCON tables compiled by the program module LOCOBLDR

LCON0104 NOT ENOUGH REGION SIZE FOR TABLE
Explanation: The member specified in the PARM to LOCOMAIN or in the @LOAD command requires a larger REGION than provided.
User Response: Increase the REGION size on the JCL EXEC statement compiled by the program module LOCOBLDR.

LCON0105 xxxx IS NOT A DEFINED SUBSYSTEM (IEFSSN..)

Explanation: The subsystem name xxxx from the PARM SSN=xxxx is not defined as a subsystem to MVS. The name xxxx must be defined in member IEFSSNxx of SYS1.PARMLIB, and must be made known to MVS by an IPL.

User Response: Check that xxxx is actually the LCON subsystem name you wish to use. Check that xxxx is defined in the current IEFSSNxx member of SYS1.PARMLIB on the cpu you are using. Check that an IPL of the cpu you are using has been performed since the subsystem name xxxx was defined in IEFSSNxx.

LCON0106 xxxxCVT IS NOT ESTABLISHED

Explanation: xxxxCVT is built by program module LOCOINIT at IPL time. Either LOCOINIT was not specified as the initialization program in IEFSSNxx for subsystem name xxxx, or it failed to run correctly at the last IPL. Program module must be available in a LNKLST library.

User Response: Check the console log at the time of the last IPL for Message xxxx700. If not found, look for other messages in the range of xxxx701 thru xxxx799 which will give information on a failure of LOCOINIT. Also look for any MVS messages related to program LOCOINIT. If no pertinent messages are found contact NETEC Technical Support at your earliest convenience.

LCON0107 FREEMAIN FAILURE

Explanation: LOCOMAIN failed to freemain a LOCOSA. LOCOMAIN will shutdown and produce some important trace information on SYSPRINT.

User Response: Retain the SYSPRINT data and contact NETEC Technical Support at your earliest convenience.

LCON0108 SUPPRESS/REROUTE TABLE GETMAIN FAILED

Explanation: There was insufficient storage available in CSA for the suppress/reroute table. Maximum size of this table is 16K.

User Response: Determine why so little CSA is available, and correct that problem.

LCON0109 SUPPRESS/REROUTE TABLE EXCEEDED (16000 BYTES)

Explanation: The suppress/reroute table is constructed in a 16000 byte area in the LOCOMAIN address space. The number of suppress/reroute statements has exceeded this size. It requires a very large number of suppress/reroute statements to exceed 16000 bytes. If you have less than 100 LCON statements with SUPPRESS or REROUTE operands, then LOCOMAIN has experienced some internal error. If you have 100 or more SUPPRESS or REROUTE operands, then you probably have legitimately exceeded the 16000 bytes.

User Response: If the message seems inappropriate, or if you would like to have a special version of LOCOMAIN with a larger suppress/reroute table, contact NETEC Technical Support.

LCON0110 EXPECTED NEW QTABLE COMMAND

Explanation: The structure of the LCON object table is incorrect. The object table may have been damaged after compilation, or LOCOBLDR may have created it in error.

User Response: Recompile the LCON table with LOCOBLDR; if the same error occurs, contact NETEC Technical Support.

LCON0111 INVALID TABLE RECORD TYPE

Explanation: The structure of the LCON object table is incorrect. The object table may have been damaged after compilation, or LOCOBLDR may have created it in error.

User Response: Insure that the release levels of LOCOBLDR and LOCOMAIN are the same. Recompile the LCON table with LOCOBLDR; if the error persists, contact NETEC Technical Support.

LCON0112 QUEUE CHAIN ERROR

Explanation: LOCOMAIN attempted to remove an LCON table statement from one of the event queues, but could not find the statement.

User Response: Contact NETEC Technical Support.

LCON0113 WAIT/POST ERROR

Explanation: LOCOMAIN was dispatched after a WAIT, but no ECB was set to X'40'.

User Response: Contact NETEC Technical Support.

LCON0114 SPECIFIED TABLE NOT FOUND

Explanation: The LCON table specified in the PARM or in the @LOAD command was not found in the LCON table library defined by JCL ddname TABLE.

User Response: Insure that the table name is spelled correctly. Insure that the library defined by the JCL ddname TABLE is the correct LCON table library, and that the requested table has been compiled by LOCOBLDR into that library.

LCON0115 INVALID SEGMENT TYPE

Explanation: The structure of the LCON object table is incorrect. The object table may have been damaged after compilation, or LOCOBLDR may have created it in error. The table may have been damaged internally by LOCOMAIN.

User Response: Insure that the release levels of LOCOBLDR and LOCOMAIN are the same. Recompile the LCON table with LOCOBLDR; if the error persists, contact NETEC Technical Support.

LCON0116 TIME QUEUE ERROR

Explanation: A statement on the TIME queue expired, but its waiting flag was not one of SLEEP, TIME, TXTIME or TOD.

User Response: Contact NETEC Technical Support.

LCON0117 ABEND AT CONSOLE REQUEST

Explanation: A @DUMP command was received from the console. A dump will be produced with USER ABEND code 1.

User Response: None.

LCON0118 COMMAND MUST BEGIN WITH x

Explanation: An LCON command was presented to LOCOMAIN from a MODIFY command. The LCON command did not start with x (the LCON command prefix character from the DEFINE statement).

User Response: LCON commands may be keyed directly on the console with no need to use the MVS MODIFY command. However, in both cases the LCON command must begin with the LCON prefix character (defaults to @).

```
LCON0119       NOT FAMILIAR WITH REQUESTED SERVICE
Explanation:   An LCON command was issued from the console or
               from an ACTION statement in the LCON table, but
               the command was not one of: LOAD, ACTIVATE,
               DEACTIVATE, STATUS, STOP, HELP, >, HOLD, RELEASE,
               RESET, TRACE, DUMP, ZERO, REFRESH,TRACE.  Message
               LCON179 show the offending command.
User Response: Correct the LCON command.

LCON0120       MUST SPECIFY MEMBER= OR TABLE=
Explanation:   An LCON @LOAD command was issued from the console
               or from an LCON ACTION statement, but the table
               was not identified by MEMBER=xxxxxxxx or
               TABLE=xxxxxxxx.
User Response: Correct the LCON command.  See the LCON USERS
               GUIDE.

LCON0121       TABLE MEMBER NOT FOUND
Explanation:   The table specified on the LOCOMAIN PARM or in the
               @LOAD command was not found in the PDS defined by
               JCL ddname TABLE.
User Response: Insure that the table name is spelled correctly.
               Insure that the library defined by the JCL ddname
               TABLE is the correct LCON table library, and that
               the requested table has been compiled by LOCOBLDR
               into that library.

LCON0122       INVALID SEGMENT LENGTH
Explanation:   The structure of the LCON object table is
               incorrect.  The object table may have been damaged
               after compilation, or LOCOBLDR may have created it
               in error.  The table may have been damaged
               internally by LOCOMAIN.
User Response: Insure that the release levels of LOCOBLDR and
               LOCOMAIN are the same.  Recompile the LCON table
               with LOCOBLDR; if the error persists, contact
               NETEC Technical Support.
```

LCON0123 INVALID QUEUE TYPE

Explanation: The structure of the LCON object table is incorrect. The object table may have been damaged after compilation, or LOCOBLDR may have created it in error. The table may have been damaged internally by LOCOMAIN.

User Response: Insure that the release levels of LOCOBLDR and LOCOMAIN are the same. Recompile the LCON table with LOCOBLDR; if the error persists, contact NETEC Technical Support.

LCON0124 INVALID REPLY NUMBER

Explanation: The reply number for a WTOR received from IEAVMXIT or from LOCOS35 is not numeric.

User Response: It is possible that a WTO text matched your WTOR table entry, in which case these is no REPLY number available.

LCON0125 xxxx NOT INITIALIZED (LOCOINIT IEFSSN..)

Explanation: LOCOMAIN was started with PARM SSN=xxxx, but the LCON subsystem named xxxx has not been initialized. LCON subsystems are initialized by LOCOINIT at IPL time, or alternately by batch execution of program LOCOINIT.

User Response: Insure that xxxx subsystem definition in your current IEFSSNxx member of SYS1.PARMLIB has LOCOINIT specified as the initialization program. Insure that your system has been IPLed since IEFSSNxx was updated.

LCON0126 INVALID WAIT FLAG (QFLAG1)

Explanation: The structure of the internal LCON table is incorrect. It has been damaged during LOCOMAIN execution.

User Response: Insure that the release levels of LOCOBLDR and LOCOMAIN are the same. Recompile the LCON table with LOCOBLDR; if the error persists, contact NETEC Technical Support.

LCON0127 INTERNAL READER NOT AVAILABLE

Explanation: LOCOMAIN attempted to open the internal reader file ddname INTRDR, but the open failed.

User Response: Insure that the JCL for LOCOMAIN has a proper definition for the INTRDR DD statement.

```
LCON0128        JOB NOT FOUND
Explanation:    A submit (@>) command was issued from the console
                or from an LCON ACTION, but the specified job was
                not found in the submit library defined by ddname
                SUBMIT.
User Response: Insure that the JCL for LOCOMAIN has a proper
                definition for the SUBMIT DD statement.  Insure
                that the jobname to be submitted was spelled
                correctly.

LCON0129        TABLE MEMBER IS NULL
Explanation:    When LOCOMAIN attempted to read the specified
                member from ddname TABLE, the member was found,
                but it had no records.
User Response: Insure that the table name is spelled correctly.
                Insure that the library defined by the JCL ddname
                TABLE is the correct LCON table library, and that
                the requested table has been compiled by LOCOBLDR
                into that library.

LCON0130        TIME OERAND REQUIRED
Explanation:    A @HOLD or @RELEASE command was issued from the
                console or from an LCON ACTION, but the TIME
                operand was not coded.  The TIME operand is
                required for consistency with other LCON commands.
User Response: Provide the required TIME operand.

LCON0131        ERROR CODE NOT KNOWN
Explanation:    Invalid code in @ERROR command.  This command is
                for testing LCON error recovery.
User response: None

LCON0132        xxxx ALREADY EXECUTING
Explanation:    LOCOMAIN attempted to start for subsystem name
                xxxx, but found that subsystem xxxx was marked
                LOCOUP.  There is apparently another LOCOMAIN in
                execution for subsystem xxxx.
User Response: If you wish to run multiple LCON subsystems, each
                must its own subsystem name.  See LCON USERS GUIDE
                for more information on running multiple LCON
                subsystems.
```

LCON0133 xxxx NOT APF AUTHORIZED

Explanation: LOCOMAIN found that it was not APF authorized, nor was an SVC number provided to obtain APF authorization.

User Response: See the LCON installation procedure in the LCON USERS GUIDE for a complete discussion of methods of providing APF authorization to LCON.

LCON0134 NOT ABLE TO OPEN SUBMIT FILE

Explanation: LOCOMAIN attempted to open the job submissiom source library ddname SUBMIT, but the open failed.

User Response: Insure that the JCL for LOCOMAIN has a proper definition for the SUBMIT DD statement.

LCON0135 xxxx TABLE NOT FOR THIS RELEASE

Explanation: LOCOMAIN found that the LCON table from the PARM or from a @LOAD command was compiled by LOCOBLDR of a different release number than the LOCOMAIN release number.

User Response: Insure that the LOCOBLDR program is the same release number as the LOCOMAIN program.

LCON0136 xxxxCVT RELEASE NUMBER NOT MATCH LOCOMAIN

Explanation: LOCOMAIN found that the xxxxCVT created by program LOCOINIT has a different release number than the LOCOMAIN release number.

User Response: Insure that the LOCOINIT program is the same release number as the LOCOMAIN program.

LCON0137 xxxx IS NOT AN APPROVED LCON SYSTEM NAME

Explanation: LOCOMAIN was requested to QUICK START an LCON with subsystem name xxxx, but xxxx is not one of the subsystem names allowed by the LCON system.

User Response: Change the SSN=xxxx parameter to specify one of the approved LCON subsystem names.

LCON0138 SSCT GETMAIN FAILED

Explanation: LOCOMAIN was requested to QUICK START an LCON and was unable to obtain the required storage for the SSCTcontrol block.

User Response: Attempt to run LOCOMAIN in a larger region. If error persists, contact NETEC Technical Support.

LCON0139 SSVT GETMAIN FAILED
Explanation: LOCOMAIN was requested to QUICK START an LCON and was unable to obtain the required storage for the SSVT control block.
User Response: Attempt to run LOCOMAIN in a larger region. If error persists, contact NETEC Technical Support.

LCON0140 SCSBRDON REQUEST TO IEAVG700 FAILED
Explanation: LOCOMAIN called IEAVG700 to request that WTOs be broadcast to all active subsystems, and the request failed.
User Response: If error persists, contact NETEC Technical Support.

LCON0141 SCSBRDOF REQUEST TO IEAVG700 FAILED
Explanation: LOCOMAIN called IEAVG700 to request that WTOs not be broadcast to all active subsystems, and the request failed.
User Response: If error persists, contact NETEC Technical Support.

LCON0142 RESUMING +>cccccccc COMMAND
Explanation: LOCOMAIN has resumed the processing of command cccccccc that was previously suspended.
User Response: none.

LCON0143 RESUMING +LOAD MEMBER=mmmmmmmm COMMAND
Explanation: LOCOMAIN has resumed the loading of table member mmmmmmmm that was presiously suspended.
User Response: none.

LCON0144 CLONE GETMAIN FAILED
Explanation: LOCOMAIN was unable to obtain the required storage for a clone copy of an LCON table entry.
User Response: Attempt to run LOCOMAIN in a larger region. If error persists, contact NETEC Technical Support.

LCON0145 CALLED PROGRAM BLOCK GETMAIN FAILED
Explanation: LOCOMAIN was unable to obtain the required storage for an internal control block used by the LCON exit program mechanism.
User Response: Attempt to run LOCOMAIN in a larger region. If error persists, contact NETEC Technical Support.

LCON0146 CALLED PROGRAM BLDL FAILED

Explanation: LOCOMAIN issued a BLDL macro for a user exit program, and the macro failed.

User Response: If error persists, contact NETEC Technical Support.

LCON0147 CALLED PROGRAM LOAD FAILED

Explanation: LOCOMAIN issued a LOAD macro for a user exit program, and the macro failed.

User Response: If error persists, contact NETEC Technical Support.

LCON0148 GETMAIN FAILED FOR SUSPEND ELEMENT, COMMAND IGNORED

Explanation: LOCOMAIN was attempting to suspend the execution of a command. The getmain for storage that is necessary to suspend the command has fails so the command has been ignored.

User Response: Reissue the command.

LCON0149 EXECUTION OF QUEUE ENTRY FAILED BECAUSE TABLE HAS BEEN RELOADED

Explanation: An attempt to execute a LOCO command through the TSO interface has been attempted. The LOCO table has been reloaded since the LOCO commands were displayed to the TSO user so the command executing is aborted.

User Response: Redisplay the queue and attempt to execute the command again.

LCON0150 CORRECT DATES AND RESUBMIT JOB

Explanation: LOCOMAIN found an error in the dates specified for the REPORT option. The error is detailed in another message.

User Response: Correct the erroneous date and resubmit the job.

LCON0151 PARM VALUE TOO LARGE, MAX SIZE = 79 BYTES

Explanation: LOCOMAIN received an execution time PARM value larger than 79 bytes.

User Response: Reduce the size of the PARM and resubmit the job.

LCON0152 DATE IN PARM HAS INCORRECT FORMAT

Explanation: LOCOMAIN received an execution time PARM for the REPORT option and found the date was not of the format MM/DD/YY as required.

User Response: Correct the erroneous date and resubmit the job.

LCON0153 MUTUALLY EXCLUSIVE DATES SPECIFIED

Explanation: LOCOMAIN received an execution time PARM for the REPORT option and found DATE specified in combination with either LODATE or HIGHDATE.

User Response: Specify either DATE parm alone or the parm pair of LODATE and HIDATE and resubmit the job.

LCON0154 NO REPORT DATE SPECIFIED

Explanation: LOCOMAIN received an execution time PARM for the REPORT option and found that none of HIDATE, LODATE or DATE parmameters were specified.

User Response: Provide date parameters and resubmit the job.

LCON0155 NO LOW REPORT DATE SPECIFIED

Explanation: LOCOMAIN received an execution time PARM for the REPORT option and found that HIDATE was specified but LODATE was not.

User Response: Provide the low date parameter and resubmit the job.

LCON0156 NO HIGH REPORT DATE SPECIFIED

Explanation: LOCOMAIN received an execution time PARM for the REPORT option and found that LODATE was specified but HIDATE was not.

User Response: Provide the high date parameter and resubmit the job.

LCON0157 LODATE / HIDATE OUT OF SEQUENCE

Explanation: LOCOMAIN received an execution time PARM for the REPORT option and found the HIDATE and LODATE values were in reverse order.

User Response: Correct the erroneous dates and resubmit the job.

LCON0158 EXIT PROGRAM HAS NEVER BEEN LOADED

Explanation: LOCOMAIN received a request to REFRESH an exit program but found that the exit program had never been loaded and need not be refreshed.

User Response: None.

LCON0159 TIMEOUT DURING SUBTASK ATTACH

Explanation: LOCOMAIN has encountered a timeout while trying to attach one of the internal subtasks.

User Response: Contact NETEC Technical support.

LCON0160 AUTHORIZATION MECHANISM HAS FAILED

Explanation: LOCOMAIN cannot get authorized.

User Response: This is probably due to an installation error. Check the installation of the authorization mechanism and correct any errors. Restart the LOCMAIN process.

LCON0161 CAFC SUPPORT IS CURRENTLY DISABLED, ACTION BYPASSED

Explanation: LOCOMAIN has attempted to execute a CAFC command and the CAFC support in LOCOMAIN is currently disabled. The action has been ignored.

User Response: Either remove the CAFC action from the LOCO table or enable the CAFC support in LOCOMAIN by placing CAFC=YES in the LOCOMAIN parm.

LCON0162 DB CONTROL SUPPORT IS CURRENTLY DISABLED, ACTION BYPASSED

Explanation: LOCOMAIN has attempted to execute a DB CONTROL command and the DB CONTROL support in LOCOMAIN is currently disabled. The action has been ignored.

User Response: Either remove the DB CONTROL action from the LOCO table or enable the DB CONTROL support in LOCOMAIN by placing DBCT=YES in the LOCOMAIN parm.

LCON0163 ATTEMPTING TO REGISTER WITH THE AUTOMATIC RESTART MANAGER

Explanation: LOCOMAIN is in the process of registering with the MVS automatic restart manager.

User Response: None.

LCON0164 REGISTRATION WITH THE AUTOMATIC RESTART MANAGER SUCCESSFUL

Explanation: LOCOMAIN has successfully registered with the MVS automatic restart manager.

User Response: None.

LCON0165 THE LOCO TABLE BEING LOADED HAS NOT BEEN SUCCESSFULLY BUILT BY LOCOBLDR

Explanation: LOCOMAIN has attempted to load a LOCO table that has not successfully completed the LOCOBLDR process.

User Response: Edit the LOCO table and correct any error. Execute the LOCOBLDR process and reload the table.

LCON0166 LCON HAS EXPERIENCED A SYNCHRONUS DCB ERROR

Explanation: LOCOMAIN has encountered a severe error while attempting to write to a CRITICAL DDNAME. LOCOMAIN is terminated.

User Response: Contact NETEC technical support.

LCON0167 REGISTRATION WITH THE AUTOMATIC RESTART MANAGER HAS FAILED RC = XXXXXXXX RS = XXXXXXXX

Explanation: LOCOMAIN has encountered an error while attempting to register with the MVS automatic restart manager.

User Response: Investigate the return and reason codes in the message.

LCON0168 CURRENT QLIMIT WOULD UTILIZE EXCESSIVE CSA, QLIMIT HAS BEEN RESET TO XXXX

Explanation: LOCOMAIN has encountered an error while processing the load of a table. LOCO has determined that the QLIMIT specified in the table would use an excessive amout of MVS CSA. LOC has automatically reset the QLIMIT to a more reasonable value.

User Response: Adjust the QLIMIT in the LOCO table.

LCON0169 WAITING FOR ACTIVE CAFC SUBTASK TO COMPLETE.

Explanation: LOCO has been ask to terminate and the CAFC subtask is busy processing a CAFCCMD command. LOCO will wait until the CAFCCMD command is complete.

User Response: None.

LCON0170 DUMP REQUESTED

Explanation: LCON has detected an internal error and is producing a memory dump with USER ABEND code 1.

User Response: If the additional error messages provide a reasonable explanation of the error, proceed with the suggested actions associated with those messages. Otherwise, contact NETEC Technical Support at your earliest convenience.

LCON0171 REISSUED WTOS IGNORED

Explanation: This is part of the STATUS command output that indicates WTOs issued by XCF for the sysplex are being ignored.

User Response: none.

LCON0172 WAITING FOR ACTIVE DBCONTROL SUBTASK TO COMPLETE.

Explanation: LOCO has been ask to terminate and the DBCONTROL subtask is busy processing a DBCTCMD command. LOCO will wait until the DBCTCMD command is complete.

User Response: None.

LCON0173 VERSION x.x (mm/dd/yyyy}

Explanation: x.x is the LCON release number; mm/dd/yyyy is the assembly of program module LOCOMAIN.

User Response: None.

LCON0174 REPLIED TO xxxx yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy

Explanation: The text yyyyyyyyyyyyyyyyyyyy was issued as a reply to reply number xxxx by LOCOMAIN.

User Response: None.

LCON0175 TABLE xxxxxxxx yyyy.ddd hh.mm.ss

Explanation: The LCON table initially loaded is xxxxxxxx. It was compiled by LOCOBLDR on date yyyy.ddd at time hh.mm.ss.

User Response: None.

LCON0176 WAITING FOR ACTIVE LOG SUBTASK TO COMPLETE.

Explanation: LOCO has been ask to terminate and the active log subtask is busy processing a logging request. LOCO will wait until the log request completes.

User Response: None.

```
WTO177   DC    0F'0',AL2(102,0)
         DC    C'----0177 CURRENT QLIMIT WOULD UTILIZE EXCESSIVE
CSA, QX
               LIMIT HAS BEEN RESET TO                      '
```

LCON0179 …………………………. Follows 119 contains bad command

LCON0180 TABLE xxxxxxxx yyyy.ddd hh.mm.ss

Explanation: The LCON table currently loaded is xxxxxxxx. It was compiled by LOCOBLDR on date yyyy.ddd at time hh.mm.ss.

User Response: None. This message produced by the @STATUS command.

LOCN0181 EVENTS WTOS WTORS TIME

Explanation: This is a heading line for message LOCO982.

User Response: None. This message produced by the @STATUS command.

LCON0182 xxxxxxxx yyyyyyyy zzzzzzzz

Explanation: This message follows LCON981 which provides headings. xxxxxxxx is the status of WTO event processing by LOCOMAIN: ACTIVE indicates WTO events are being processed, it is set by @ACTIVATE WTO. PAUSED indicates WTO events are not being processed, it is set by @DEACTIVATE WTO. yyyyyyyy is the status of WTOR event processing by LOCOMAIN: ACTIVE indicates WTOR events are being processed, it is set by @ACTIVATE WTOR. PAUSED indicates WTOR events are not being processed, it is set by @DEACTIVATE WTOR. yyyyyyyy is the status of TIME event processing by LOCOMAIN: ACTIVIVE indicates TIME events are being processed, it is set from paused status by @ACTIVATE TIME, it is set from held status by @RELEASE TIME. PAUSED indicates TIME events are not being processed, it is set by @DEACTIVATE TIME. HELD indicates TIME events are being held for later processing after @RELEASE TIME, it is set by @HOLD TIME.

User Response: None. This message produced by the @STATUS command.

LCON0183 TRANSFER SLOTS USED xxxxx OF yyyyy

Explanation: There are yyyyy pointer slots in LOCOCVT used to locate LOCOSA control blocks. The slots are used dynamically as LOCOSAs are allocated and freed. xxxxx is the high water mark of slot usage.

User Response: None. This message produced by the @STATUS command.

LCON0184 MESSAGES LOST (QUEUE FULL): xxxxxx

Explanation: xxxxxx Count of WTOs have been lost to LCON due to QLIMIT overflow.

User Response: Consider increasing CLIMIT (on DEFINE card). It is possible that LOCOMAIN is dispatching too slowly.

LCON0185 LAST ACTIVITY AT hh:mm:ss
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
Explanation: The last ACTION performed by LCON was xxxxxxxxxxxxxxxx at time of day hh:mm:ss.
User Response: None. This message produced by the @STATUS command.

LCON0186 WTORS   SEEN xxxxxx
Explanation: The number of WTOR events seen by LOCOMAIN is xxxxxx.
User Response: None. This message produced by the @STATUS command.

LCON0187 WTOS   SEEN xxxxxx
Explanation: The number of WTO events seen by LOCOMAIN is xxxxxx.
User Response: None. This message produced by the @STATUS command.

LCON0188 COMMANDS SEEN xxxxxx
Explanation: The number of LCON commands seen by LOCOMAIN is xxxxxx.
User Response: None. This message produced by the @STATUS command.

LCON0189 ACTIONS TAKEN xxxxxx
Explanation: The number of LCON ACTIONs processed by LOCOMAIN is xxxxxx.
User Response: None. This message produced by the @STATUS command.

LCON0190 REPLYS GIVEN xxxxxx
Explanation: The number of REPLYs issued by LOCOMAIN is xxxxxx.
User Response: None. This message produced by the @STATUS command.

LCON0191 OK
Explanation: Okay.
User Response: None.

LCON0192 ====> xxxxxxxxxxxx
Explanation: This message is preceded by LCON993. xxxxxxxxxx is an available LCON console command.
User Response: None. This message is produced in response to @HELP.

LCON0193 LCON CONSOLE COMMANDS ARE

Explanation: This message is a heading for messages LCON992.

User Response: None. This message is produced in response to @HELP.

LCON0194 ABENDS RECOVERED (10 MAXIMUM): xxxxxx

Explanation: LOCOMAIN has abended xxxxxx times and recovered.

User Response: Ignore unless pervasive, then notify NETEC. LCON trace may point to problems.

LCON0195 LCON CALLED PROGRAM PPPPPPPP WAS IN CONTROL

Explanation: LCON user written exit program named PPPPPPPP was in control at the time of an abend.

User Response: Fix user written exit program error.

LCON0196 EXECUTION ABORTED

Explanation: LCON has detected an internal error and is terminating. There should be additional messages describing the error.

User Response: If the additional error messages provide a reasonable explanation of the error, proceed with the suggested actions associated with those messages. Otherwise, contact NETEC Technical Support at your earliest convenience.

LCON0197 LCON HAS EXPERIENCED AN ABEND

Explanation: The LCON ESTAE exit is in control due to an unexpected condition.

User Response: Save the dump, and notify NETEC Technical Support at your earliest convenience.

LCON0198 LCON IS ATTEMPTING TO WITHDRAW FROM THE SYSTEM

Explanation: The LCON ESTAE exit is in control due to an unexpected condition. LOCOMAIN is beginning a controlled shutdown.

User Response: None.

LCON0199 LCON HAS WITHDRAWN FROM SYSTEM SUCCESSFULLY

Explanation: The LCON ESTAE exit is in control due to an unexpected condition. LOCOMAIN has completed a controlled shutdown.

User Response: None.

Messages from LOCOINTF (LCON0601-LCON0699)

Messages are shown with LCON as the message prefix. This prefix will be changed to the current subsystem name by LOCOINTF at execution time. If LOCOINTF is run on behalf of subsystem LCOB, then the messages will be LCOB500, LCOB501, etc.

PARM VALUE TOO LARGE, MAX SIZE = 79 BYTES
Explanation: Parm value is too large. It may not be longer than 79 bytes.
User Response: Remove extraneous data from PARM.

LCON0601 PARM MSG NOT BEGIN WITH QUOTE
Explanation: Parm MSG= did not have a quote after the = sign.
User Response: Provide the required quotes around the MSG text.
Example: PARM='SSN=xxxx,MSG=''message'''

LCON0602 MSG PARM TOO LONG
Explanation: The text of the MSG PARM was longer than 80 bytes.
User Response: Shorten the MSG text to no more than 80 bytes.

LCON0603 LOCOCVT NOT SET UP
Explanation: The subsystem name in the PARM SSN=xxxx does not have a LOCOCVT established (initialized).
User Response: Insure that LOCOINIT has been run properly for this subsystem name.

LCON0604 SUBSYSTEM NOT FOUND
Explanation: The PARM SSN=xxxx specified a subsystem name unknown to MVS. xxxx defaults to LCON.
User Response: Insure that xxxx is the proper subsystem name for this LCON. Insure that xxxx is defined in IEFSSN in the current SYS1.PARMLIB, and that the system has been IPLed since xxxx was defined in IEFSSN.

LCON0605 NOT ABLE TO OPEN SYSPRINT
Explanation: The open for ddname SYSPRINT failed.
User Response: Insure that there is JCL for ddname SYSPRINT.

LCON0606 xxxx NOT RUNNING
Explanation: The LOCOCVT for LCON subsystem name xxxx is marked LOCODOWN. No LOCOMAIN is active for subsystem name xxxx.
User Response: Insure that LOCOMAIN is running for the specified subsystem name before executing LOCOINTF.

LCON0607 xxxx STOPPED RUNNING
Explanation: The LOCOCVT for LCON subsystem name xxxx was marked LOCODOWN while LOCOINTF was in execution.
User Response: Insure that LOCOMAIN is running for the specified subsystem name before executing LOCOINTF.

LCON0608 xxxx LOCOSA GETMAIN FAILED
Explanation: LOCOINTF was unable to obtain CSA storage for a LOCOSA.
User Response: Determine why CSA storage was not available.

LCON0609 XM POST FAILED
Explanation: LOCOINTF was unable to POST LOCOMAIN. LOCOMAIN may have terminated.
User Response: Contact NETEC Technical Support at your earliest convenience.

LCON0610 SAQ ERROR
Explanation: LOCOINTF attempted to delete a LOCOSA pointer from the SAQ, but could not find the proper slot.
User Response: Contact NETEC Technical Support at your earliest convenience.

LCON0611 NOT AUTHORIZED
Explanation: LOCOINTF found that it was not APF authorized, nor was an SVC number provided to obtain APF authorization.
User Response: See the LCON installation procedure in the LCON USERS GUIDE for a complete discussion of methods of providing APF authorization to LCON.

LCON0612 BOX QUEUE FULL
Explanation: In attempting to pass a pseudo-WTO to LOCOMAIN, LOCOINTF found that the queue was full. This can happen when LOCOMAIN has a relatively low dispatching priority.
User Response: Increase the dispatching priority of LOCOMAIN.

Messages from LOCO Subtask (LCON0700-LCON0799)

LCON0700 CAFC INTERFACE SUBTASK BEGINNING INITIALIZATION
Explanation: This is an information message indicating the LOCOCAFC subtask is beginning initialization.
Issued by: LOCOCAFC
User Response: none.

LCON0701 CAFC INTERFACE SUBTASK TRACE ACTIVATED
Explanation: This is an information message indicating the LOCOCAFC subtask has activated internal trace.
Issued by: LOCOCAFC
User Response: none.

LCON0703 CAFC INTERFACE SUBTASK TRACE DEACTIVATED
Explanation: This is an information message indicating the LOCOCAFC subtask has stopped internal tracing.
Issued by: LOCOCAFC
User Response: none.

LCON0704 CAFC INTERFACE SUBTASK BEGINNING TERMINATION
Explanation: This is an information message indicating the LOCOCAFC subtask is beginning termination.
Issued by: LOCOCAFC
User Response: none.

LCON0706 CAFC INTERFACE SUBTASK INITIALIZATION COMPLETE
Explanation: This is an information message indicating the LOCOCAFC subtask has completed initialization.
Issued by: LOCOCAFC
User Response: none.

LCON0707 CAFC INTERFACE SUBTASK TERMINATION COMPLETE
Explanation: This is an information message indicating the LOCOCAFC subtask has completed termination.
Issued by: LOCOCAFC
User Response: none.

LCON0708 ERROR ALLOCATING CAFC CONTROL FILE a.b.c.d
Explanation: The LOCO CAFC subtask has attempted to allocate the CAFC control file a.b.c.d. The allocation has failed.
Issued by: LOCOCAFC
User Response: Correct the control file name in the LOCO table CAFCCMD command.

LCON0709 CAFC COMMAND ABORTED

Explanation: This is an information message indicating the previous CAFC command has been aborted.

Issued by: LOCOCAFC

User Response: Determine the original error and correct.

LCON0710 UNABLE TO LOAD CAFC PROGRAM AFCP2016

Explanation: The LOCO CAFC subtask has attempted to load the CAFC batch interface program AFCP2016. AFCP2016 was not found in the STEPLIB concatenation.

Issued by: LOCOCAFC

User Response: Add the library containing the AFCP2016 program to the LOCO STEPLIB concatenation.

LCON0712 CAFC COMMAND COMPLETED RC= xx

Explanation: This is an information message indicating the LOCOCAFC subtask has completed the execution of a CAFC command with return code xx.

Issued by: LOCOCAFC

User Response: none.

LCON0713 EXECUTING CAFC COMMAND xxxxxxxx

Explanation: This is an information message indicating the LOCOCAFC subtask is executing command xxxxxxxx.

Issued by: LOCOCAFC

User Response: none.

LCON0714 TERMINATING EXECUTION OF CAFCCMD DUE TO FORCE REQUEST

Explanation: This is an information message indicating the LOCOCAFC subtask has interrupted a current execution of a CAFCCMD due to an operator action that required a LOAD, FORCE of the current LOCO table.

Issued by: LOCOCAFC

User Response: none.

LCON0715 REMOVING CAFC QUEUE ENTRY FROM THE LOCOCAFC WORK QUEUE DUE TO PREVIOUS FORCE

Explanation: This is an information message indicating the LOCOCAFC subtask has removed pending CAFC actions due to n operator action that required a LOAD, FORCE of the current LOCO table.

Issued by: LOCOCAFC

User Response: none.

LCON0720 DB CONTROL INTERFACE SUBTASK BEGINNING INITIALIZATION
Explanation: This is an information message indicating the LOCO DB Control subtask is beginning initialization.
Issued by: LOCODBCT
User Response: none.

LCON0721 DB CONTROL INTERFACE SUBTASK TRACE ACTIVATED
Explanation: This is an information message indicating the LOCO DB Control subtask has activated internal trace.
Issued by: LOCODBCT
User Response: none.

LCON0723 DB CONTROL INTERFACE SUBTASK TRACE DEACTIVATED
Explanation: This is an information message indicating the LOCO DB Control subtask has stopped internal tracing.
Issued by: LOCODBCT
User Response: none.

LCON0724 DB CONTROL INTERFACE SUBTASK BEGINNING TERMINATION
Explanation: This is an information message indicating the LOCO DB Control subtask is beginning termination.
Issued by: LOCODBCT
User Response: none.

LCON0726 DB CONTROL INTERFACE SUBTASK INITIALIZATION COMPLETE
Explanation: This is an information message indicating the LOCO DB Control subtask has completed initialization.
Issued by: LOCODBCT
User Response: none.

LCON0727 DB CONTROL INTERFACE SUBTASK TERMINATION COMPLETE
Explanation: This is an information message indicating the LOCO DB Control subtask has completed termination.
Issued by: LOCODBCT
User Response: none.

LCON0729 DB CONTROL COMMAND ABORTED
Explanation: This is an information message indicating the previous DB control command has been aborted.
Issued by: LOCODBCT
User Response: Determine the original error and correct.

LCON0730 UNABLE TO LOAD DB CONTROL PROGRAM DBC2000
Explanation: The LOCO DB Control subtask has attempted to load the DB Control batch interface program DBC2000. DBC2000 was not found in the STEPLIB concatenation.
Issued by: LOCODBCT
User Response: Add the library containing the DBC2000 program to the LOCO STEPLIB concatenation.

LCON0731 INVALID LOCO QTABLE ENTRY RECEIVED BY LOCODBCT
Explanation: The LOCO DB Control subtask has encountered a DBCTCMD that did not contain an IMSID keyword.
Issued by: LOCODBCT
User Response: Correct the DBCTCMD command and reload the LOCO table.

LCON0732 DB CONTROL COMMAND COMPLETED RC=
Explanation: This is an information message indicating the LOCO DB Control subtask is executing command xxxxxxxx.
Issued by: LOCODBCT
User Response: none.

LCON0733 EXECUTING DB CONTROL COMMAND xxxxxxxx
Explanation: This is an information message indicating the LOCO DB Control subtask is executing command xxxxxxxx.
Issued by: LOCODBCT
User Response: none.

LCON0734 TERMINATING EXECUTION OF DBCTCMD DUE TO FORCE REQUEST
Explanation: This is an information message indicating the LOCO DB Control subtask has interrupted a current execution of a DBCTCMD due to an operator action that required a LOAD, FORCE of the current LOCO table.
Issued by: LOCODBCT
User Response: none.

LCON0735 REMOVING DBCT QUEUE ENTRY FROM THE LOCODBCT WORK QUEUE DUE TO PREVIOUS FORCE
Explanation: This is an information message indicating the LOCO DB Control subtask has removed pending DB Control actions due to an operator action that required a LOAD, FORCE of the current LOCO table.
Issued by: LOCODBCT
User Response: none.

LCON0740 LIBRARY COMPRESSION SUBTASK BEGINNING INITIALIZATION
Explanation: This is an information message indicating the library compression subtask is beginning initialization.
Issued by: LOCOCPRS
User Response: none.

LCON0741 LIBRARY COMPRESSION SUBTASK TRACE ACTIVATED
Explanation: This is an information message indicating the library compression subtask has activated internal trace.
Issued by: LOCOCPRS
User Response: none.

LCON0743 LIBRARY COMPRESSION SUBTASK TRACE DEACTIVATED
Explanation: This is an information message indicating the library compression subtask has stopped internal tracing.
Issued by: LOCOCPRS
User Response: none.

LCON0744 LIBRARY COMPRESSION SUBTASK BEGINNING TERMINATION
Explanation: This is an information message indicating the library compression subtask is beginning termination.
Issued by: LOCOCPRS
User Response: none.

LCON0746 LIBRARY COMPRESSION SUBTASK INITIALIZATION COMPLETE
Explanation: This is an information message indicating the library compression subtask has completed initialization.
Issued by: LOCOALOG
User Response: none.

LCON0747 LIBRARY COMPRESSION SUBTASK TERMINATION COMPLETE
Explanation: This is an information message indicating the library compression subtask has completed termination.
Issued by: LOCOCPRS
User Response: none.

LCON0748 LIBRARY COMPRESSION SUBTASK DYNAMIC ALLOCATION /DEALLOCATION ERROR

Explanation: This is an information message indicating the library compression subtask has encountered an error while attempting to allocate or de-allocate the library to compress.

Issued by: LOCOCPRS

User Response: Correct the reason the library could not be allocated or de-allocated.

LCON0749 LIBRARY COMPRESSION ABORTED

Explanation: This is an information message indicating the compression of a LOCO library has failed. Check previous messages for additional information.

Issued by: LOCOCPRS

User Response: none.

LCON0752 LIBRARY COMPRESSION COMPLETED RC= xx

Explanation: This is an information message indicating the compression of a LOCO library has completed with return code xx.

Issued by: LOCOCPRS

User Response: none.

LCON0753 COMPRESSING a.b.c.d

Explanation: This is an information message indicating the LOCO library, a.b.c.d is being compressed.

Issued by: LOCOCPRS

User Response: none.

LCON0760 LOCO ACTIVITY LOG SUBTASK BEGINNING INITIALIZATION

Explanation: This is an information message indicating the LOCO activity log subtask is beginning initialization.

Issued by: LOCOALOG

User Response: none.

LCON0761 LOCO ACTIVITY LOG SUBTASK TRACE ACTIVATED

Explanation: This is an information message indicating the LOCO activity log subtask has activated internal trace.

Issued by: LOCOALOG

User Response: none.

LCON0763 LOCO ACTIVITY LOG SUBTASK TRACE DEACTIVATED

Explanation: This is an information message indicating the LOCO activity log subtask has stopped internal tracing.

Issued by: LOCOALOG

User Response: none.

LCON0764 LOCO ACTIVITY LOG SUBTASK BEGINNING TERMINATION
Explanation: This is an information message indicating the LOCO activity log subtask is beginning termination.
Issued by: LOCOALOG
User Response: none.

LCON0766 LOCO ACTIVITY LOG SUBTASK INITIALIZATION COMPLETE
Explanation: This is an information message indicating the LOCO activity log subtask has completed initialization.
Issued by: LOCOALOG
User Response: none.

LCON0767 LOCO ACTIVITY LOG SUBTASK TERMINATION COMPLETE
Explanation: This is an information message indicating the LOCO activity log subtask has completed termination.
Issued by: LOCOALOG
User Response: none.

Messages from LOCOINIT (LCON0800-LCON0899)

Messages are shown with LCON as the message prefix. This prefix will be changed to the current subsystem name by LOCOINIT at execution time. If LOCOINIT is run on behalf of subsystem LCOG, then the messages will be LCOG800, LCOG801, etc.

LCON0800 xxxx VERSION x.x (mm/dd/yyyy) INITIALIZED
Explanation: This is a confirmation that LCON subsystem xxxx has been has been initialized by LOCOINIT release x.x which was assembled on mm/dd/yyyy.
User Response: None.

LCON0801 xxxx NOT DEFINED AS A SUBSYSTEM
Explanation: The PARM SSN=xxxx specified a subsystem name unknown to MVS. xxxx defaults to LCON.
User Response: Insure that xxxx is the proper subsystem name for this LOCO. Insure that xxxx is defined in IEFSSN in the current SYS1.PARMLIB, and that the system has been IPLed since xxxx was defined in IEFSSN.

LCON0802 xxxx ALREADY INITIALIZED
Explanation: The PARM SSN=xxxx specified a subsystem name which has already been initialized by either LOCOINIT.
User Response: If you wish to re-initialize an LCON subsystem, then you must first un-initialize it with program LOCOINIT using PARM='SSN=xxxx,REVERSE'.

LCON0803 GETMAIN FOR LOCOCVT FAILED
Explanation: LOCOINIT was unable to obtain X'13CC' bytes of CSA storage. LOCOMAIN will not be able to run.
User Response: Contact NETEC Technical Support at your earliest convenience.

LCON0804 xxxx NOT YET INITIALIZED
Explanation: LCON Subsystem xxxx has not been initialized, and therefore cannot be un-initialized with PARM='REVERSE'.
User Response: It is possible that LOCOINIT failed for subsystem xxxx at the previous IPL. Insure that xxxx is the subsystem name which you wish to un-initialize.

LCON0805 xxxx FREEMAIN FOR LOCOCVT FAILED
Explanation: LOCOINIT was unable to freemain the LOCOCVT for subsystem name xxxx.
User Response: Contact NETEC Technical Support at your earliest convenience.

LCON0806 xxxx MUST BE TERMINATED BEFORE REVERSE-INIT
Explanation: LOCOINIT found that LOCOCVT for subsystem name xxxx was marked LOCOUP. There is a LOCOMAIN currently in execution for subsystem name xxxx.
User Response: If you really wish to un-initialize LCON subsystem xxxx, then terminate the LOCOMAIN which is using it.

LCON0807 xxxx VERSION x.x (mm/dd/yyyy) REVERSE-INIT
Explanation: This is a confirmation that LCON subsystem xxxx has been has been un-initialized by LOCOINIT release x.x which was assembled on mm/dd/yyyy.
User Response: None.

LCON0810 PARM VALUE TOO LARGE, MAX SIZE = 79 BYTES
Explanation: Parm value is too large. It may not be longer than 79 bytes.
User Response: Remove extraneous data from PARM.

LCON0811 GETMAIN FOR SSCT EXTENSION FAILED
Explanation: An internal error has occurred.
User Response: Contact NETEC Technical support.

LCON0812 SUBSYSTEM WAS FOUND INITIALIZED WITH LOCO RELEASE x.x
Explanation: An attempt has been made to initialize the LOCO subsystem but the subsystem interface is already initialized with LOCO release x.x.
User Response: Run a REVERSE initialization to deactivate the current subsystem initialization and then attempt to reinitialize the subsystem.

LCON0813 THE SUBSYSTEM IS INACTIVE, REVERSE INIT IS REQUIRED
Explanation: The LOCO subsystem that who is being initialized is not currently active. A reverse initialization can be executed to remove the LOCO subsystem.
User Response: Execute a reverse initialization.

LCON0814 SUBSYSTEM IS ACTIVE

Explanation: The LOCO subsystem that is being initial is currently active.

User Response: Stop the current LOCO subsystem and then attempt to initialize.

LCON0815 FAILURE OPENING STEPLIB DDNAME

Explanation: The LOCO subsystem has attempted to open the STEPLIB DDNAME. An error has occurred during this process.

User Response: Correct the error and restart LOCO.

LCON0816 BLDL FAILED TRYING TO LOCATE LOCOSSSM

Explanation: The LOCO subsystem cannot find the LOCOSSSM module in the STEPLIB DDNAME.

User Response: Place the library that contains the LOCOSSSM program into the LOCOMAIN STEPLIB concatenation.

LCON0817 GETMAIN FAILED FOR RESIDENT LOCOSSSM

Explanation: The LOCO subsystem has attempted to obtain storage for the LOCOSSSM program and the GETMAIN for storage has failed.

User Response: Determine why the GETMAINED failed and restart LOCOMAIN.

LCON0818 LOCOSSSM LOAD FAILED

Explanation: The LOCO subsystem has attempted to load the LOCOSSSM program and an error has occurred.

User Response: Determine why the load failed and restart LOCOMAIN.

LCON0820 SIZE OF MODULE IN BLDL LIST IS ZERO

Explanation: The LOCO subsystem found the size of the LOCOSSSM module to be zero in the BLDL entry for the module.

User Response: Contact NETEC Technical Support.

LCON0821 LOAD OF THE LOCOSSSM MODULE HAS FAILED

Explanation: The LOCO subsystem has attempted to load the LOCOSSSM module and the load has failed.

User Response: Determine why the load failed and restart LOCOMAIN.

Messages from LOCOIMEX (LCON0900-LCON0999)

LCON0000I LOCO TABLE IMPORT/EXPORT UTILITY IS A PROPRIETARY PROGRAM FROM NETEC INTERNATIONAL, COPYRIGHT 2004
Explanation: This is an informational message.
User Response: None.

LCON0900I LOCO TABLE IMPORT/EXPORT UTILITY BEGINNING EXECUTION AT HH:MM:SS ON MM/DD/YYYY
THIS PROGRAM IS EXECUTING DURING A TRIAL PERIOD THAT WILL EXPIRE ON XX/DD/YYYY
THE TRIAL PERIOD FOR EVALUATING THIS PROGRAM HAS EXPIRED, EXECUTION TERMINATING.
Explanation: This is an informational message indicating the date and time the LOCO table import/export utility is executing. If the product is executing during a trial period the second message will be displayed informing the user of the trial period. If the product is executing under a trial period and the trial period has expired the third message will be displayed and the execution of the LOCO table import/export utility will terminate.
User Response: None.

LCON0902E INVALID DATA ENTERED IN THE PARM FIELD, DATA IGNORED
Explanation: The LOCO import/export utility has found invalid data in the parm field on the EXEC statement.
User Response: Remove or correct the information and resubmit the job for execution.

LCON0903E TSO ENVIRONMENT SETUP FAILED RC = XX
Explanation: An internal error has occurred during the initialization phase of the LOCO import/export utility while attempting to setup the batch TSO environment.
User Response: Contact NETEC technical support.

LCON0904I READY FOR COMMAND INPUT
Explanation: This is an informational message.
User Response: None.

LCON0905E INVALID COMMAND IN PREVIOUS CONTROL CARD
Explanation: The LOCO import/export utility has found an invalid control card in the previous control card statement.
User Response: Correct the control card and submit the job for execution.

LCON0907I COMMAND COMPLETE RC = XX
Explanation: This is an informational message indicating the previous LOCO import/export utility command has completed with the return code XX.
User Response: If RC=00 then the command completed successfully. If RC not equal 00 investigate why the command failed.

LCON0909E PARSE ERROR IN PREVIOUS COMMAND
Explanation: The previous LOCO import/export utility control card contains a parse error.
User Response: Correct the error and submit the job for execution.

LCON0910E THE MEMBER NAME SPECIFIED IS INVALID
Explanation: The member name specified on the LOCO utility import/export utility control card is invalid.
User Response: Correct the error and submit the job for execution.

LCON0911E INVALID REPLACE OPTION ON THE MEMBER KEYWORD
Explanation: The replace option on the MEMBER keyword is invalid.
User Response: Correct the error and submit the job for execution.

LCON0913E TSO STACK SETUP FAILED RC = XX
Explanation: An internal error has occurred during the initialization phase of the LOCO import/export utility while attempting to setup the batch TSO stack.
User Response: Contact NETEC technical support.

LCON0914E INDD AND INDSN ARE MUTUALLY EXCLUSIVE
Explanation: The LOCO import/export utility has found both the INDD and INDSN keywords on an IMPORT control statement. These keywords are mutually exclusive and one must be removed.
User Response: Correct the error and submit the job for execution.

LCON0915E EITHER INDD OR INDSN IS REQUIRED
Explanation: Neither the INDD or INDSN keyword was found on the LOCO import/export utility IMPORT control card. One of the keywords is required.
User Response: Correct the error and submit the job for execution.

LCON0916E INDD MISSING FROM THE JCL
Explanation: The user has specified the INDD key word on an IMPORT control card and the DDNAME specified in the INDD keyword is not found in the JCL for the job.
User Response: Correct the error and submit the job for execution.

LCON0917E EITHER TABLEDD OR TABLEDSN IS REQUIRED
Explanation: Neither the TABLEDD or TABLEDSN keyword was found on the LOCO import/export utility IMPORT control card. One of the keywords is required.
User Response: Correct the error and submit the job for execution.

LCON0918E TABLEDD AND TABLEDSN ARE MUTUALLY EXCLUSIVE
Explanation: The LOCO import/export utility has found both the TABLEDD and TABLEDSN keywords on an IMPORT control statement. These keywords are mutually exclusive and one must be removed.
User Response: Correct the error and submit the job for execution.

LCON0919E TABLEDD MISSING FROM THE JCL
Explanation: The user has specified the TABLEDD key word on an IMPORT control card and the DDNAME specified in the TABLEDD keyword is not found in the JCL for the job.
User Response: Correct the error and submit the job for execution.

LCON0920E THE MEMBER KEYWORD IS REQUIRED
Explanation: The member keyword is required on the LOCO import/export utility IMPORT control card.
User Response: Correct the error and submit the job for execution.

LCON0921E THE MEMBER CAN NOT BE FOUND
Explanation: The member specified in the MEMBER keyword of the LOCO import/export utility IMPORT control card cannot be found.
User Response: Correct the error and submit the job for execution.

LCON0922E THE INDSN CAN NOT BE FOUND

Explanation: The user has specified the INDSN key word on an IMPORT control card and the data set name specified in the INDSN keyword is not be found.

User Response: Correct the error and submit the job for execution.

LCON0923E THE TABLEDSN CAN NOT BE FOUND

Explanation: The user has specified the TABLEDSN key word on an IMPORT control card and the data set name specified in the TABLEDSN keyword is not be found.

User Response: Correct the error and submit the job for execution.

LCON0924E OUTDD AND OUTDSN ARE MUTUALLY EXCLUSIVE

Explanation: The LOCO import/export utility has found both the OUTDD and OUTDSN keywords on an EXPORT control statement. These keywords are mutually exclusive and one must be removed.

User Response: Correct the error and submit the job for execution.

LCON0925E EITHER OUTDD OR OUTDSN IS REQUIRED

Explanation: Neither the OUTDD or OUTDSN keyword was found on the LOCO EXPORT utility control card. One of the keywords is required.

User Response: Correct the error and submit the job for execution.

LCON0926E OUTDD MISSING FROM THE JCL

Explanation: The user has specified the OUTDD key word on an EXPORT control card and the DDNAME specified in the OUTDD keyword is not found in the JCL for the job.

User Response: Correct the error and submit the job for execution.

LCON0927E EITHER TABLEDD OR TABLEDSN IS REQUIRED

Explanation: Neither the TABLEDD or TABLEDSN keyword was found on the LOCO import/export utility EXPORT control card. One of the keywords is required.

User Response: Correct the error and submit the job for execution.

LCON0928E TABLEDD AND TABLEDSN ARE MUTUALLY EXCLUSIVE

Explanation: The LOCO import/export utility has found both the TABLEDD and TABLEDSN keywords on an EXPORT control statement. These keywords are mutually exclusive and one must be removed.

User Response: Correct the error and submit the job for execution.

LCON0929E TABLEDD MISSING FROM THE JCL

Explanation: The user has specified the TABLEDD key word on an EXPORT control card and the DDNAME specified in the TABLEDD keyword is not found in the JCL for the job.

User Response: Correct the error and submit the job for execution.

LCON0930E THE MEMBER KEYWORD IS REQUIRED

Explanation: The member keyword is required on the LOCO import/export utility EXPORT control card.

User Response: Correct the error and submit the job for execution.

LCON0931E THE MEMBER CAN NOT BE FOUND

Explanation: The member specified in the MEMBER keyword of the LOCO import/export utility EXPORT control card is not found.

User Response: Correct the error and submit the job for execution.

LCON0932E THE OUTDSN CAN NOT BE FOUND

Explanation: The user has specified the OUTDSN key word on an EXPORT control card and the data set name specified in the OUTDSN keyword is not be found.

User Response: Correct the error and submit the job for execution.

LCON0933E THE TABLEDSN CAN NOT BE FOUND

Explanation: The user has specified the TABLEDSN key word on an EXPORT control card and the data set name specified in the TABLEDSN keyword is not be found.

User Response: Correct the error and submit the job for execution.

LCON0934E THE TABLEDSN A.B.C.D MUST BE A PDS
Explanation: The LOCO table data set specified either via a TABLEDD or TABLEDSN keyword must be a PDS. Inspection of the data set has found otherwise.
User Response: Correct the error and submit the job for execution.

LCON0935E THE TABLEDSN A.B.C.D IS NOT A LOCO TABLE LIBRARY
Explanation: The LOCO table data set specified either via a TABLEDD or TABLEDSN must be the format of a LOCO table library. Inspection of the data set has found otherwise.
User Response: Correct the error and submit the job for execution.

LCON0937E REPLACE IS INVALID FOR SEQUENTIAL OUTPUT
Explanation: An EXPORT command is being processed whose output is a sequential file. The REPLACE option has been specified on the MEMBER keyword. This is invalid.
User Response: Correct the error and submit the job for execution.

LCON0938E THE LRECL OF THE OUTDD DATASET MUST BE 80
Explanation: The output data set of a LOCO EXPORT command must have an LRECL=80. Inspection of the data set specified by either the OUTDD or OUTDSN key word has found otherwise.
User Response: Correct the error and submit the job for execution.

LCON0939E NEW MEMBER NAME NOT ALLOWED WHEN GENERIC MEMBER NAME SPECIFIED
Explanation: The MEMBER KEYWORD of a LOCO EXPORT contains a generic member name. The new member name has been specified. The new member name cannot be specified when the member name is generic.
User Response: Correct the error and submit the job for execution.

LCON0940E EXPECTED CONTINUATION NOT RECEIVED
Explanation: While processing a LOCO import/export control card, an expected continuation was not received
User Response: Correct the error and submit the job for execution.

LCON0941E LOCO TABLE IMPORT/EXPORT UTILITY DYNAMIC ALLOCATION ERROR

$\left\{\begin{matrix}\text{ALLOCATE}\\\text{DEALLOCATE}\end{matrix}\right\}$ DSNAME=a.b.c.d MEMBER=mmmmmmmm RC=rrrrrrrr

Explanation: The LOCO import/export utility has encountered a dynamic allocation error attempting to allocate or deallocate the dsname a.b.c.d. The return code is rrrrrrrr

User Response: Correct the error and submit the job for execution.

LCON0942I MEMBER mmmmmmmm $\left\{\begin{matrix}\text{ADDED}\\\text{REPLACED}\end{matrix}\right\}$

Explanation: This is an informational message indicating the member mmmmmmmm has either been added or replaced.

User Response: None.

LCON0943E MEMBER mmmmmmmm NOT $\left\{\begin{matrix}\text{ADDED}\\\text{REPLACED}\end{matrix}\right\}$ BECAUSE NO SPACE LEFT IN THE DIRECTORY

Explanation: The member mmmmmmmm has not been added or replaced because there was no space left in the PDS directory.

User Response: Correct the error and submit the job for execution.

LCON0944E MEMBER NOT ADDED BECAUSE OF I/O ERROR

Explanation: The member was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0945E MEMBER NOT ADDED BECAUSE DCB NOT OPEN

Explanation: The member was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0946E MEMBER NOT ADDED BECAUSE OF INSUFFICIENT VIRTUAL STORAGE

Explanation: The member was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0947E ALIAS NOT ADDED BECAUSE OF INVALID TTR

Explanation: The alias was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0948E MEMBERS LEFT IN PENDING DELETE STATE
Explanation: The members was left in a pending delete state because of an internal error.
User Response: Contact NETEC technical support.

LCON0949E THE LRECL OF THE INDD DATASET MUST BE 80
Explanation: The logical record length of the dataset specified either by INDD or INDSN on the IMPORT control card is not 80.
User Response: Correct the error and submit the job for execution.

LCON0950E MEMBER KEYWORD IS INVALID FOR SEQUENTIAL INPUT
Explanation: The MEMBER keyword was found on a LOCO import/export utility IMPORT control card when the INDD or INDSN indicated a sequential data set. In this situation the sequential input must contain ./ ADD and ./ REPL control cards.
User Response: Correct the error and submit the job for execution.

LCON0951E NEW MEMBER NAME NOT ALLOWED WHEN GENERIC MEMBER NAME SPECIFIED
Explanation: The MEMBER KEYWORD of a LOCO IMPORT contains a generic member name. The new member name has been specified. The new member name cannot be specified when the member name is generic.
User Response: Correct the error and submit the job for execution.

LCON0952I ALIAS aaaaaaaa $\left\{\begin{matrix}\text{ADDED}\\\text{REPLACED}\end{matrix}\right\}$
Explanation: This is an information message indicating the alias has been added or replaced.
User Response: None.

LCON0953E ALIAS aaaaaaaa NOT $\left\{\begin{matrix}\text{ADDED}\\\text{REPLACED}\end{matrix}\right\}$ BECAUSE NO SPACE LEFT IN THE DIRECTORY
Explanation: The alias name has not been added or replaced because there is no space left in the PDS directory.
User Response: None.

LCON0954E ALIAS NOT ADDED BECAUSE OF I/O ERROR
Explanation: The alias was not added because of an internal error.
User Response: Contact NETEC technical support.

LCON0955E ALIAS NOT ADDED BECAUSE DCB NOT OPEN

Explanation: The alias was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0956E ALIAS NOT ADDED BECAUSE OF INSUFFICIENT VIRTUAL STORAGE

Explanation: The alias was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0957E ALIAS NOT ADDED BECAUSE OF INVALID TTR

Explanation: The alias was not added because of an internal error.

User Response: Contact NETEC technical support.

LCON0958E ALIAS LEFT IN PENDING DELETE STATE

Explanation: The alias was left in a pending delete state because of an internal error.

User Response: Contact NETEC technical support.

LCON0959E INVALID IEBUPDTE CONTROL CARD ENCOUNTERED

Explanation: An invalid ./ ADD or ./ REPL control card has been found in the sequential input to a LOCO import/export utility IMPORT command.

User Response: Correct the error and submit the job for execution.

LCON0960E IEBUPDTE CONTROL CARD MISSING

Explanation: A LOCO import/export IMPORT command is being processed with sequential input from the INDD or INDSN keyword. The input must contain valid ./ ADD or ./ REPL control cards. No such cards were encountered.

User Response: Correct the error and submit the job for execution.

LCON0961E MEMBER mmmmmmmm NOT IMPORTED, MEMBER WAS FOUND AND $\left\{\begin{array}{l}\text{./ REPL}\\ \text{REPLACE}\end{array}\right\}$ WAS NOT SPECIFIED

Explanation: The member mmmmmmmm was not imported because member mmmmmmmm was found in the TABLEDD or TABLEDSN and replace was not specified either via a ./ REPL control card or the REPLACE option on the MEMBER keyword.

User Response: Correct the error and submit the job for execution.

LCON0962I INVOKING LOCO TABLE BUILDER FOR MEMBER mmmmmmmm

Explanation: This is an informational message indicating the BUILD(YES) option is active on a LOCO import/export utility IMPORT command and the LOCOBLDR has been invoked to build the table for member mmmmmmmm.

User Response: None.

LCON0963I LOCO TABLE BUILD PROCESS WAS $\left\{\begin{array}{c}\text{SUCCESSFUL}\\\text{UNSUCCESSFUL}\end{array}\right\}$ SUCCESSFUL/UNSUCCESSFUL FOR MEMBER mmmmmmmm RC = XX

Explanation: This is an informational message indicating the success or failure of the LOCOBLDR process. The LOCOBLDR programs return code is XX.

User Response: None.

LCON0964E MEMBER mmmmmmmm NOT ADDED/REPLACED BECAUSE OF PREVIOUS ERROR

Explanation: There has been a previous error that has prevented the adding or replacing of the member mmmmmmmm.

User Response: Correct the error and submit the job for execution.

LCON0998E LOCO TABLE IMPORT/EXPORT UTILITY STOPPED DUE TO PREVIOUS ERRORS

Explanation: This is an informational message indicating the LOCO import/export utility has terminated due to errors.

User Response: Correct the error and submit the job for execution.

LCON0999I LOCO TABLE IMPORT/EXPORT UTILITY EXECUTION COMPLETE RC = XX

Explanation: This is informational message indicating that this execution of the LOCO import/export utility is complete with the highest return code encountered XX.

User Response: None.