
Unicenter

NetMaster Socket Management for CICS Administrator Guide

Version 1.0



Computer Associates
The Software That Manages eBusiness



This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

Chapter 1: Introduction

The Unicenter NetMaster Interface	1-1
Unicenter NetMaster Interface Command Processor	1-2
The Listen Tool	1-3
Tracing	1-4
The Socket Management Environment	1-5

Chapter 2: Operations

Initialization	2-2
Socket Management Initialization Messages	2-3
Initialization Failure	2-4
Socket Management Termination	2-5
Termination Messages	2-6
TCP Transport Provider Failure	2-6
Transaction Security	2-7

Chapter 3: T09CONxx Customization

Configuration Table Suffixing	3-1
Customizing the Translation Table	3-3

Chapter 4: T09CONxx Configuration Reference

Socket Management Definition Statements	4-2
T09MCICS Macro	4-2
Changing Translation Tables	4-8
T09MLSTN	4-8
Client/Data Listener Option	4-10
Client/Data Option Data Structure	4-12

Examples	4-12
T09MCMDS	4-13
T09MTRAN	4-15
Parameter List Passed to T09MTRAN Transactions	4-19
T09MEND	4-20

Chapter 5: CICS/TS Resource Definition Reference

System Initialization Tasks (SIT) Entries	5-2
Program List Table (PLT) Entries	5-3
Sample CICS/TS RDO Member T09RDO	5-4
Transient Data Queue (TDQ) Entries	5-7
Program Entries	5-8
Transaction Entries	5-10
Terminal Control (TCT) Entries	5-12

Chapter 6: Unicenter NetMaster Command Processor Security

Security Configuration for eTrust CA-ACF2	6-2
Security Configuration for eTrust CA-Top Secret	6-3
Security Configuration for RACF	6-4

Chapter 7: Diagnostic Commands

TCPEEP	7-1
User Interface	7-2
Trace Data Collected	7-3
Viewing Trace Data	7-3
Trace Operation	7-3
TCPEEP Syntax	7-4
Useful TCPEEP Commands	7-9
TCPEEP Example	7-10
TRACE	7-12
MVS Component Trace	7-12
Trace Address Space	7-12
Exit	7-13
External Writer	7-13
Component Trace JCL	7-13
TRACE Command	7-14

TRACE Command Reply	7-16
TRACE Command Reply Options	7-17
Comparing TCPEEP and MVS TRACE Syntax	7-20
Trace Command Examples	7-21
Processing Trace Data	7-23

Chapter 8: Troubleshooting

Diagnostic Procedures	8-2
Collecting Diagnostic Data	8-2
Debugging Guidelines for a Socket Management Application	8-3
Interpreting Diagnostic Data	8-10
Accessing the Online Client Support System	8-11
Requirements for Using StarTCC	8-11
StarTCC Security	8-12
Accessing StarTCC	8-12
Accessing the Technical Support Phone Services Directory	8-13
CA-TLC: Total License Care	8-14
Contacting Technical Support	8-14
Generating a Problem Report	8-15
CAISERV Utility	8-15
Product Versions and Maintenance	8-18
Requesting Enhancements	8-18

Appendix A: Security Program

About Optional Security Program	A-1
Security Program Logic Flow	A-2
Security Program Return	A-2
The Security Communications Block	A-3

Appendix B: Unicenter NetMaster Administrator Interface

Starting the Interface	B-1
Moving Through the Panels	B-1
Socket Management Query Display (SQ)	B-2
Socket Management and CPT Bounce (SB)	B-2
Socket Management CMD Server Bounce (SSB)	B-3
CICS Server Start (SS)	B-3
CICS Transaction Start (TS)	B-4

IP CICS Connections (ICC)	B-4
Connections List (CL)	B-5
Socket Management Initialization and Customization Services Parameter (admin function)	B-7
Signon Details	B-8

Appendix C: The Terminal Administrator Interface

Entering the Administrator Interface	C-2
Function Codes and PF Keys	C-2
Administrator Interface Navigation	C-3
Exiting the Administrator Interface	C-4
Bypassing the Main Menu	C-4
Utilization	C-5
Utilization Summary Panel	C-5
Summary Statistics	C-7
List of Socket Management Endpoints	C-7
Displaying Inactive Endpoints	C-9
Explicit Selection of the Detail Display	C-9
Browsing Detail for Client or Server Task	C-10
Generating an Online Trace Entry	C-10
Client/Server Browse Detail Panel	C-10
Examples	C-11
Configuration	C-17
Configuration Table Parameters Panel	C-18
Operations	C-21
Operations Control Menu Panel	C-22
Socket Management Interface Error Log Panel	C-23
Quiesce Idle Listeners Function	C-25
Drain Socket Management Interface Function	C-25
Shutdown All Immediate Function	C-26
Reset All Statistics Function	C-26
Startup Socket Management Interface Function	C-27

Index

Introduction

This chapter provides an overview of Unicenter NetMaster Socket Management for CICS (Socket Management).

It discusses the following topics:

- [The Unicenter NetMaster Interface](#) – Describes the interface and integration of CICS TCP sockets information into Unicenter NetMaster for TCP/IP.management suite
- [The Listen Tool](#) – Briefly describes the listen tool provided with Socket Management for establishing TCP/IP connections
- [Tracing](#) – Briefly describes the tracing facilities available for application programs in CICS to communicate with a TCP/IP network
- [The Socket Management Environment](#) – Briefly describes the installation and customization that must be done to make Socket Management available

The Unicenter NetMaster Interface

Unicenter NetMaster for TCP/IP (Unicenter NetMaster) acts as a central repository for network information in an OS/390 enterprise environment. By enabling this interface all Socket Management endpoints for a sysplex can be managed from a single point.

Once the Unicenter NetMaster interface is enabled, Socket Management does the following:

- Passes additional CICS/TS information about TCP endpoints to Unicenter NetMaster such as User ID, CICS/TS transaction name, and CICS/TS transaction number.
- All EZASOKET and EZACICAL calls are time stamped at entry and exit. You can tell whether a transaction is executing inside CICS or API code. You can tell how long a call has been active.
- EZASOKET and EZACICAL call statistics are kept at both the session and server level. These session and server level statistics are available for query.

- This information then becomes available via central network management displays within Unicenter NetMaster, refer to the Unicenter NetMaster documentation for additional details.
- Provides an interface, the Unicenter NetMaster command processor allows further drill down inquiries into Socket Management endpoints.

Note: Unicenter NetMaster 6.2 requires:

- Service pack 3 (GL0206) installed
- Place statement 'PROD=SOCKETMGMT' in the region RUNSYSIN file to enable the Socket Management Interface

Unicenter NetMaster Interface Command Processor

The Unicenter NetMaster Interface Command Processor is enabled when you code something on the parameter PORT= in the T09MCMDS macro in the configuration file T09CONez. See the "Configuration Statement Reference" chapter for full details.

Note: It is strongly recommended you turn this feature on by coding something in the T09MCMDS macro.

The Command Processor, a socket-based CICS application, facilitates *administrative connections* between Unicenter NetMaster and Unicenter NetMaster Socket Management for CICS. These connections monitor Socket Management EZASocket endpoint activity and initiate other CICS/TS services.

Unicenter NetMaster clients who successfully connect to the Socket Management command processor can have the following commands issued for them to a CICS/TS Socket Management region.

Note: In most cases you are not aware of which command is being issued for you. From the Unicenter NetMaster perspective, all you see is the drill down via a GUI or menu driven screen. The reason these commands are pointed out is that a reference to the command may occur in logging if a problem occurs. In the rare case that you drill down to issuing line mode commands in Unicenter NetMaster, the following is a list of the supported command set you will see:

CONNDrop	Drop a CPT connection or listener.
CONNGet	Display detailed information about an individual CPT connection.
CONNList	Display a list of CPT connections.
CPTBounce	Restart the CPT interface.
CPTStatus	Display the overall status of CPT.

SMSRVRBounce	Restart the command server (T09TCMDS).
StartServer	Start a CICS listener/server.
StartTransaction	Start a CICS transaction.

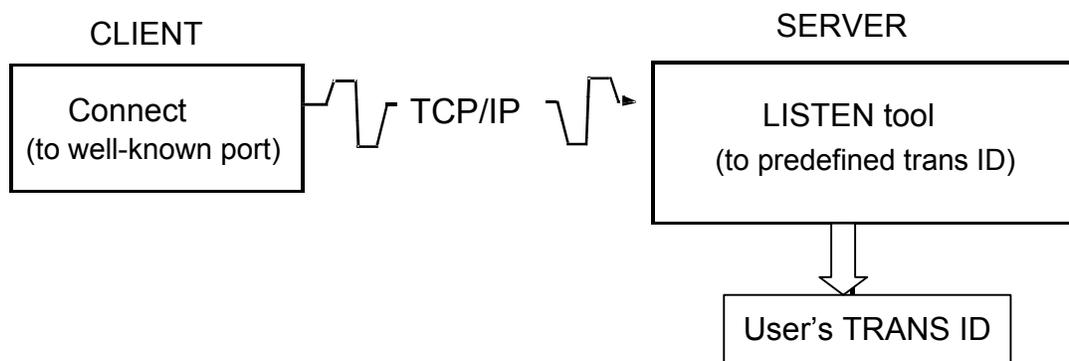
The command processor's well-known port number is transmitted to the Clients can then request connections to the command server across that port. Once a client is verified (assuming command server security has not been disabled), you can then drilldown and use menu options that issue the above commands. This enables you to monitor or disable socket connections in the command server CICS region, or issue commands to start other CICS transactions and servers.

The Listen Tool

The Socket Management Listen tool is comprised of prewritten CICS transactions that help establish connections over Transmission Control Protocol (TCP). You can add a new listener service in your environment using two to three parameters of a macro definition.

The T09MLSN macro starts the IPTL server transaction, which replaces the CSKL listener transaction.

Note: There is no limit to the number of listen tools you can start. To customize another listen tool just code another T09MLSTN macro in the T09CONez configuration table.



Tracing

The Socket Management API and tools provide extensive debugging and performance analysis capabilities through its tracing and statistics options. The debugging capabilities can be invaluable in an environment where problems can occur in the CICS/TS application, such as the TCPIP address space, the network, or the remote host. The tracing options used selectively can pinpoint exactly where a problem is occurring.

You can follow EZASOCKET and EZASOCKET calls and their parameter lists both into and out of the TCP/IP region.

You can follow TCP/IP sessions as their ownership passes between CICS tasks.

All trace entries use time stamps that detail events down to the micro second level.

Tracing is turned on and off dynamically – invoked via the TCPEEP interface. There is no need to modify CICS transactions or configuration files. Just submit a TCPEEP job with the options you want traced and the trace events are routed to the trace address space.

All trace output is routed to a trace address space. There is no CICS overhead when formatting trace output. Tracing does not overload the CICS message logs.

The Socket Management Environment

Installation and customization must be performed before Socket Management is available. Installation consists of allocating the Socket Management data sets and executing SMP/E. Customization requires a configuration file and CICS/TS resource definitions.

The configuration file defines the operating environment and automated Socket Management tools transactions. The environment information enables Socket Management to communicate with both CICS and the transport provider's API. The automated tools transactions are optional.

CICS/TS resource definitions define programs, transaction IDs, and transient data queues. Resource definition statements for Socket Management operating environment programs and transactions are required; while those for Socket Management tools, transient data queues are optional.

Socket Management can be enabled and disabled without shutting down CICS/TS.

You can also:

- Apply maintenance
- Load new configurations, or the transport provider can terminate and reinitialize without CICS/TS being recycled

Operations

This chapter provides operation information for Unicenter NetMaster Socket Management for CICS.

It includes the following topics:

- [Initialization](#) – Describes the methods available for initializing the Socket Management Interface and for diagnosing initialization failure
- [Socket Management Termination](#) – Describes the methods available for terminating the Socket Management Interface and for diagnosing termination failure
- [Transaction Security](#) – Describes how to secure Socket Management initialization and termination transaction to prevent unauthorized use

The initialization and termination can be executed automatically by using the CICS/TS Program List Tables (PLT). The Interface can be started and stopped manually, any number of times, without the need to bring down the CICS/TS system. This capability to reinitialize the Interface allows a new configuration table module to be loaded at anytime.

Initialization

The Unicenter NetMaster Socket Management for CICS Interface is initialized by program, T09TSTRT, which has a default transaction ID of IPST. A different transaction ID can be configured in T09MCICS.

Invoke this program using one of the following methods to initialize the Socket Management Interface:

- Specify the T09TSTRT program as an entry in the CICS/TS PLT table for program initialization (PLTPI). This automatically starts up the Socket Management Interface as part of CICS/TS initialization. This process is described in the Configuration Table Suffixing section.
- Enter the IPST EZ transaction from a CICS/TS terminal. Where, EZ is the suffix of your configuration table. This manual start up of the Socket Management Interface is useful on test systems, where the Interface can be brought up and down manually to test a new Configuration Table. If a new copy of the T09CONEZ configuration module was assembled, then the T09TSTRT program automatically loads the newest version of that load module from its DFHRPL load library concatenation. This process is described fully in the Configuration Table Suffixing section in the chapter “Installation and Configuration.”

Regardless of which startup method you use, the T09TSTRT program performs the same set of tasks to initialize the Socket Management environment:

- Enables Task-Related User Exit programs, T09COMON and EZACIC01.
- Loads the Configuration Table module, T09CONFIG, or one of the alternative configuration tables of format T09CONEZ that is associated with a region, as explained in the Configuration Table Suffixing section of the chapter “Installation and Configuration.”
- Opens the Transient Data queues for Error, Statistics, and Trace (the default names are ACER, ACST, and ACTR, respectively).
- Connects to the address space of the TCP transport provider. The TCP/IP started task name is specified by the JOBNAME= *parm* in the Configuration Table.
- Resolves the local hostname.
- Loads the Translation Table module.
The default name is T09XENG.
- Starts all the Listener tools specified in the Configuration Table.
- Starts the Unicenter NetMaster interface command processor if specified in the Configuration Table.

Socket Management Initialization Messages

When the previous initialization tasks are accomplished, startup messages are written out to the system console log. The following set of messages indicates a successful initialization:

```
JOB21989 +T09116I T09TSTRT CONFIGURATION TABLE T09CONez HAS BEEN LOADED
JOB21989 +T09180I T09TSTRT STARTING Unicenter NetMaster Socket Management for CICS 1.0.0
JOB21989 +T09183I T09TSTRT LMP Code=2D, STARTRAK=NSOCCS Abbreviation is Unicenter NetMaster
JOB21989 +T09100I T09TSTRT EZACIC01 TRUE EXIT INTERFACE ENABLED
JOB21989 +T09124I T09CINIT CPT ESTABLISHED SOCKETS COMPATIBILITY WITH JOBNAME: tcip
JOB21989 +T09111I T09CINIT CPT DEFAULT TRANSLATION TABLE T09XENG LOADED
JOB21989 +T09147I T09CINIT CA-NetMaster Network management exit successfully enabled
JOB21989 +T09181I T09TSTRT INITIALIZATION SUCCESSFUL FOR Unicenter NetMaster Socket Management for CICS
1.0.0
```

Note: The job name *tcip* shown in these messages is the default started task name for the transport provider, and may be different for your installation.

A set of messages is also written to the Socket Management Transient Data queues at initialization. In most cases, this output goes to the MSGUSR DD; otherwise, the messages go to wherever the TDQ ACER is redirected within the CICS/TS region. If all three queues are routed to the same CICS/TS sysout data set, then these messages appear together in the CICS/TS sysout to indicate a successful initialization:

```
00029 19:02:51 T09123I T09CINIT INITIAL WRITE TO ERROR LOG TD QUEUE
00029 19:02:51 T09810I T09CINIT INITIAL WRITE TO STATISTICS LOG TD QUEUE
00029 19:02:51 T09936I T09CINIT INITIAL WRITE TO TRACE LOG TD QUEUE
00029 19:02:51 T09124I T09CINIT ESTABLISHED SOCKETS COMPATIBILITY WITH JOBNAME: tcip
00029 19:02:51 T09111I T09CINIT DEFAULT TRANSLATION TABLE T09XENG LOADED
00029 19:02:51 T09147I T09CINIT CA-NetMaster Network management exit successfully enabled
00035 19:02:51 T09493I T09TCMDS SocketView command server started on port 2257
```

The 00029 preceding these messages is the CICS/TS task number of the IPST startup transaction performing the initialization.

Initialization Failure

If a Socket Management/Tool transaction or a user-written application using the Socket Management/API services gets an AEY9 abend, it is an indication that:

- The Socket Management Interface was not started, or
- The initialization has failed.

Note: AEY9 is a standard CICS/TS abend issued when an application call is made to a task-related user exit that is not enabled.

There are several reasons why Socket Management initialization might fail. First, the TCP address space be up and running with its API active when Socket Management initialization is started. It must be the TCP transport provider specified by the `JOBNAME= parm` of the current configuration table. If the TCP API is not available, Socket Management initialization fails and the following messages are sent to the console log:

```
+T09116I T09TSTRT CONFIGURATION TABLE T09CONEz HAS BEEN LOADED
+T09180I T09TSTRT STARTING Unicenter NetMaster Socket Management for CICS 1.0.0
+T09183I T09TSTRT LMP Code=2D,STARTRAK=NSOCCS Abbreviation is Unicenter NetMaster
+T09100I T09TSTRT EZACIC01 TRUE EXIT INTERFACE ENABLED
+T09102I T09CINIT PRODUCT INITIALIZATION PENDING API CONNECTION
+T09129E T09CINIT UNABLE TO INITIALIZE API FOR JOBNAME tcpip, ERRNO=10102
+T09105E T09CINIT INITIALIZATION FAILED - CHECK CICS LOG FOR DETAILS
+T09114E T09TSTRT INITIALIZATION ROUTINE FAILED RC= 18
+T09132E T09TSTRT EZACIC01 TRUE EXIT INTERFACE DISABLED
+T09105E T09TSTRT INITIALIZATION FAILED - CHECK CICS LOG FOR DETAILS
```

The *tcpip* shown in message T09129E is the `JOBNAME= parm` in the T09MCICS configuration table entry.

Another reason for initialization failure is if the Task-Related User Exit (TRUE) cannot be enabled. This occurs if CICS/TS cannot load the TRU Exit modules, T09COMON or EZACIC01, or the configuration table module, T09CONEz, or the translation table module, T09XENG, or user-provided alternate translation table. This can occur if the actual load module is missing from a DFHRPL load library, or if a problem exists with its CICS/TS RDO definition, such as having the module currently marked DISABLED.

Finally, initialization can fail if any of the three Transient Data queues specified by the `Q NAMES= parm` in the configuration table are not defined in the CICS/TS RDO definitions or cannot be opened. In all of these cases, error messages are sent to the console syslog, ending with the T09105E...INITIALIZATION FAILED message.

Socket Management Termination

The Interface is terminated by program, T09TTERM, which has a default transaction ID of IPPR. This program can be invoked in the following ways to shutdown the Interface:

- Specify the T09TTERM program as an entry in the CICS/TS PLT table for system shutdown (PLTSD).

This automatically shuts down the Socket Management Interface as part of CICS/TS termination.

This process is described in the Configuration Table Suffixing section of the chapter "Installation and Configuration."

- Enter the IPPR transaction from a CICS/TS terminal.

This manual shutdown of the Socket Management Interface is useful on test systems, perhaps to load a new copy of the configuration table module, T09CONez.

- Enter the CPTbounce command from the NetMaster command processor interface.

Regardless of how the shutdown program (T09TTERM) is invoked, it performs the same set of tasks to terminate the Socket Management environment:

- Issues an abortive close on all TCP and UDP endpoints still active
- Disconnects from the address space of the TCP transport provider, whose tcpip started task name is specified by the JOBNAME= *parm* in the configuration table
- Releases the Translation Table module, T09XENG
- Disables the Task-Related User Exit programs, T09COMON and EZACIC01
- Releases the configuration table module, T09CONez

Termination Messages

As these tasks are accomplished, termination messages are written out to the system console log.

The following messages indicate a successful termination:

```
19.16.45 JOB21989 +T09182I T09CTERM SHUTDOWN COMPLETE FOR Unicenter NetMaster Socket Management for CICS
1.0.0
19.16.45 JOB21989 +T09132E T09TTERM EZACIC01 TRUE EXIT INTERFACE DISABLED
19.16.45 JOB21989 +T09120I T09TTERM TERMINATION SUCCESSFUL
```

There is also a set of messages written out to the Transient Data queues at Socket Management termination. In most cases, this output goes to the MSGUSR DD, otherwise the messages go wherever the TDQ ACER is redirected to within the CICS/TS region.

Note: The number of messages prior to T09901I may vary depending on the number of endpoints that are forced to close at termination.

```
00054 18:30:49 T09901I MODULE T09TTERM 03/22/02 18.27 C6006761 ENTERED AT 18:30:49 2002.081
00054 18:30:49 T09106I T09CTERM API ENDPOINT 156BB008 CLOSED
00054 18:30:49 T09106I T09CTERM API ENDPOINT 157B5008 CLOSED
00054 18:30:56 T09130I T09TTERM PROGRAMMER INTERFACE DISABLED
00054 18:30:56 T09131I T09TTERM T09CONEz CONFIGURATION TABLE RELEASED
```

The 000nn preceding these messages is the CICS/TS task number of the CICS/TS transaction on whose behalf these Socket Management messages were issued.

TCP Transport Provider Failure

If the address space for the TCP transport provider fails or is shut down, the Socket Management termination program, T09TTERM, is automatically invoked. All application endpoints are abortively closed, as if the IPPR transaction had been executed. The messages sent to the console log and TD queues are the same as when the termination program is invoked by transaction or the Administrator Interface.

It is the responsibility of the application program to handle the return codes for the immediate disconnects that could be returned from its last Socket Management call, indicating that the Socket Management Interface is terminating. The application should rollback any committed resources if it has a syncpoint established. The Socket Management/Tools do this automatically when their Transient Data queues are protected. The application should also be ready to handle the AEY9 abend that occurs if the application tries to issue a Socket Management call when the Interface is down.

Transaction Security

Access to the Unicenter NetMaster Socket Management for CICS startup and shutdown functions can be restricted by using normal CICS/TS transaction security. Regardless of whether an external security manager or native CICS/TS security is used, the following CICS/TS transactions should be protected:

- IPST – Socket Management Interface Initialization
- IPPR – Socket Management Interface Termination
- IPAO – Administrator Interface Operations Control

If native CICS/TS security is used, this is done by specifying a `TRANSEC= value` greater than one in the CICS/TS RDO definition for these transactions. This requires that a terminal user be signed on with that transaction security key before he can execute the start up or shut down functions.

T09CONxx Customization

This chapter provides information for customizing Unicenter NetMaster Socket Management for CICS (Socket Management).

It includes these sections:

- [Configuration Table Suffixing](#) – Describes how to give alternate names to your configuration table
- [Customizing the Translation Table](#) – Describes how to modify or create new translation tables using the SMP/E USERMOD facility

Note: Other customization information is contained in the “T09CONxx Configuration Reference” and “CICS/TS Resource Definition Reference” chapters of this guide. Software installation instructions are available in *Getting Started*.

Configuration Table Suffixing

The default configuration table name is T09CONFIG. We recommend that sites configure, assemble and link a T09CONFIG member into their T09LOAD or table library. Simply copy the sample member T09CONEZ from T09SAMP library and rename it to T09CONFIG. When a site uses the default T09CONFIG configuration table name, it makes product start up simpler. You can:

- Start the product with the IPST transaction (associated with program T09TSTRT) without any parameters
- Place program T09TSTRT in the PLTPI table without having to code a SIT INITPARM override for the proper configuration parameter override

Multiple configuration tables can reside in a load library. You can link edit the tables with different characters in either the last single position or the last two positions of the configuration table name. This allows multiple CICS/TS regions on one or many hosts to have unique LISTEN tools or NetMaster interfaces defined. Each single-position suffix or two-character position suffix can be specified to the startup transaction IPST when starting Socket Management from a terminal. The suffix can also be passed to the startup program T09TSTRT through a CICS/TS SIT table's INITPARM parameter at CICS/TS initialization time.

A suffix specified dynamically from a CICS/TS terminal takes priority over any INITPARM option.

The suffix is entered as a parameter to the startup transaction:

```
IPST X1
```

Loads configuration table T09CONX1.

Secondary to a CICS/TS terminal start up of Socket Management is starting via a PLTPI entry for the T09TSTRT program. See chapter "CICS/TS Resource Definition Reference" for more details on the PLT entries. The default product configuration table is T09CONFIG. A site that runs the product with a non-default T09CONxx configuration file may pass the two-digit suffix in the CICS/TS SIT table INITPARM parameter overrides. For example, place the following INITPARM entry in the SIT overrides to start the product using the T09CONA2 configuration table during CICS startup:

```
INITPARM=(T09TSTRT='A2')
```

Loads configuration table T09CONA2.

If you use a configuration table other than T09CONFIG, it must be defined to CICS/TS RDO definitions as a program as the following example shows.

```
CEDA DEF PROG(T09CONA2) LANG(ASSEMBLER) G(T09CPT)
```

If neither override method is used, the default table name, T09CONFIG, is loaded.

Customizing the Translation Table

The ability to modify or create new translation tables is controlled through the SMP/E USERMOD facility. To incorporate a modified or new translation table into Socket Management, receive and apply the modules to SMP/E CSI.

It is recommended that sites do not modify the default or distributed translation table named T09XENG. Distributed maintenance could regress site modifications. Update and submit member T09UTRAN in your CNTL library to apply a new translation table.

Additional translation tables are in the T09SAMP library. Make a backup copy of the translation table to your T09SAMP library, modify, and run the following USERMOD:

```
//<NAME> JOB (000), 'TRANS USER MOD'
/*
/*      SAMPLE JCL TO RECEIVE AND APPLY USERMOD TO CHANGE
/*      THE CPT TRANSLATION TABLE (T09XENG).
/*
/*      CHANGE 'XXX' TO THE CORRECT CPT FMID.
/*
/*
//SMPE      EXEC PGM=GIMSMP,REGION=4096K,TIME=960,
//          PARM= 'CSI=SMPINDX.CSI,PROCESS=WAIT'
//SMPHOLD   DD DUMMY
//SMPLOG    DD DSN=SMPINDX.SMPLOG,DISP=MOD
//SMPDOUT   DD SYSOUT=*
//SMPPTFIN  DD *
++ USERMOD (MU1TRAN) .
++ VER (Z038)
          FMID(T090xxx) /* CHANGE TO CPT FMID */ .
++ SRC (T09XENG) TXLIB(T09SAMP) DISTMOD(AT09LOAD)DISTLIB(AT09SAMP) .
/*
//SMPCNTL  DD *
          SET BDY(GLOBAL) .
          RECEIVE S(MU1TRAN) .
          SET BDY(TCPTZN) .
          APPLY S(MU1TRAN) .
/*
```


T09CONxx Configuration Reference

This chapter details how to configure the Socket Management environment and the automated tools. The environmental macro instructions are required, while the automated tool macro instructions are optional.

Note: See *Getting Started* for general software installation instructions.

- The T09MCICS macro instruction (required) defines the Socket Management environment. This macro defines the communication subsystem, support transaction IDs, default translation table, and log transient data queues to the CICS/TS interface routines.
- You select the LISTEN tool by defining the T09MLSTN macro statement. Each TCP server port can be defined, along with its associated data processing transaction ID. The LISTEN tool can be defined to let the client determine either the transaction ID to start or the transient data queue in which to write client-specified data.
- The T09MCMDS macro defines the Unicenter NetMaster Interface for passing commands from Unicenter NetMaster to Unicenter NetMaster Socket Management for CICS
- The T09END macro statement (required) defines termination of the configuration. It does not need parameter settings. It is an environmental configuration macro instruction.

These macro statements are assembled and link edited into a load module called T09CONez and placed in the Socket Management load library.

- The T09CONez member in the T09SAMP library contains the distributed configuration table
- The T09ASMLK member in the CNTL library contains JCL to assemble and link edit the configuration table

Note: The last two characters of the configuration name can be any suffix. See the Configuration Table Suffixing section in the “T09CONxx Customization” chapter for instructions on how to accomplish this.

Socket Management Definition Statements

This section describes the Socket Management macros and the parameters used to define the Socket Management environment and tools.

T09MCICS Macro

Defines the associated TCP communication subsystem, the CICS/TS transaction IDs, and queue names used by the Socket Management interface.

Important! *There must be only one of these macros in the configuration, and it must be the first statement.*

```
T09MCICS
[ AIPREFIX = ccc ]
[ , AISTATOP = YES | NO ]
[ , AITABSZ = 1000 ]
[ , AITDSTAT = dest ]
[ , AITSELOG = num_storage_recs ]
[ , AITSTRAC = num_temp_stor_recs ]
[ , CICSENT = Y | N ]
[ , JOBNAME = tcpstack ]
[ , LCAFTRAN = IPLF ]
[ , LCASIZE = 4072 ]
[ , LCAFTRAN = IPLF ]
[ , LCASIZE = 4072 ]
[ , LINGER = seconds ]
[ , MSOCK = max_concurrent_sockets ]
[ , MTAKE = seconds ]
[ , QLSTN = backlog_queue_num ]
[ , QNAMES = ( stat, trace, error ) ]
[ , RETRYINT = seconds ]
[ , RETRYMAX = num_retries ]
[ , SCTYEXIT= ] NORMAL | MANDTORY ]
[ , SCKTVIEW=Yes | No ]
[ , SSN=subsysname ]
[ , TRANSID = ( stop, [, |stn ] [, strt ] [, lst2 ] ]
[ , TRANSTBL = tblname ]
[ , TRCSSN = trace_address_space_subsystem ]
[ , USERID = userid ]
[ , USRTRNID = IPUL ]
```

AIPREFIX

(Administrator Interface Transid Prefix). Three-character prefix of all of the Administrator Interface transactions to avoid conflict with existing transaction IDs. Also defines the temporary storage queue names used by the Administrator Interface.

Default: IPA.

AISTATOP	<p>(Administrator Interface Statistics Option). Defines if Socket Management should collect cumulative statistics for online display by the Administrator Interface.</p> <p>Default: No.</p>
AITABSZ	<p>(Administrator Interface Table Size). Number of table entries used to hold Socket Management cumulative statistics for online display.</p> <p>Default: 1000.</p>
AITDSTAT	<p>(Administrator Interface Statistics Destination). An optional transient data destination to which statistics are written when Socket Management is shut down, or when statistics capture is reset.</p> <p>Default: Null – No queue name defined.</p>
AITSELOG	<p>(Administrator Interface Temporary Storage for Error Log). Number of temporary storage records to hold Error Log entries for online display.</p> <p>The maximum is 999. Zero disables the Error Log display.</p> <p>Default: 30.</p> <p>Note: When this value is nonzero and error messages are logged, one transaction per message (IPAI) is started in CICS.</p>
AITSTRAC	<p>(Administrator Interface Temporary Storage for Trace). Number of temporary storage records to hold online trace entries.</p> <p>The maximum is 999. Zero disables the online trace.</p> <p>Note: When this value is nonzero and trace messages are being logged, one transaction per trace entry (IPAI) is started in CICS.</p>
CICSENT	<p>Determines whether a EXEC CICS ENTER TRACE command executes every time a Socket Management service routine is entered.</p> <p>Y=On N=Off</p> <p>Default: Yes.</p>

JOBNAME	<p>Required. Job name or step name of the TCP stack to use for IBM CICS sockets (EZASOKET/EZACICAL) endpoints. This is the job name of either a TCP/IP stack or Unicenter TCPAccess stack.</p> <p>Default: "TCPIP".</p> <p>Note: In order to use sockets, you must define a JOBNAME.</p>
LCAFTRAN	<p>Transaction ID for the program T09TLCAF. It is used to free LCA storage at product shutdown.</p> <p>Default: IPLF.</p>
LCASIZE	<p>LCASIZE below the line storage allocate for use as a LCA control block. Each server that runs requires 12 bytes of LCA storage. Additionally, there is a 12 byte header for the LCA storage.</p> <p>Valid values: 0-32767.</p> <p>Default: 612 decimal.</p>
LINGER	<p>Seconds to wait on a TCP orderly close call to complete. This is the reception of a FIN-ACK at the TCP transport layer from the remote host in response to a FIN(finish) initiated from the local connection endpoint. A FIN is created due to a shutdown socket API call.</p> <p>Default: One.</p>
MSOCK	<p>Maximum number of concurrent sockets per initapi socket call that Socket Management supports. Typically, each connection requires a socket. In general, every server and every individual transaction calls initapi or has one implied for them by doing a socket function requiring an initapi.</p> <p>Note: Only one initapi call is implied per CICS/TS transaction number.</p> <p>Maximum: 2000.</p> <p>Default: 50.</p>

MTAKE	<p>Maximum number of seconds to wait for a spawned CICS/TS task (transaction) to takesocket a connection. If the timeout occurs before the spawned task does a takesocket call, the connection is closed, only if the TS queue created with the EXEC CICS START FROM option no longer exists. That is an EXEC CICS RETRIEVE call was done by the new task or the task exited.</p> <p>Note: Listen Backlog Queue is used on all LISTEN service request calls unless overridden by <i>setsockopt</i>.</p> <p>Default: Zero.</p>						
QLSTN	<p>Listen Backlog Queue used on all LISTEN service request calls unless overridden by <i>setsockopt</i>.</p> <p>Default: Five.</p>						
QNAMEs	<p>Transient data queue names for various classes of messages produced by the Socket Management routines. Entries must be defined in the CICS/TS RDO for each of the queue names specified, or Socket Management initialization will fail.</p> <table><tr><td><i>stat</i></td><td><p>Transient data queue name where statistics information is logged.</p><p>Default: ACST.</p></td></tr><tr><td><i>trace</i></td><td><p>Transient data queue name where trace information is logged.</p><p>Default: ACTR.</p></td></tr><tr><td><i>error</i></td><td><p>Transient data queue name where error messages information is logged.</p><p>Default: ACER.</p></td></tr></table>	<i>stat</i>	<p>Transient data queue name where statistics information is logged.</p> <p>Default: ACST.</p>	<i>trace</i>	<p>Transient data queue name where trace information is logged.</p> <p>Default: ACTR.</p>	<i>error</i>	<p>Transient data queue name where error messages information is logged.</p> <p>Default: ACER.</p>
<i>stat</i>	<p>Transient data queue name where statistics information is logged.</p> <p>Default: ACST.</p>						
<i>trace</i>	<p>Transient data queue name where trace information is logged.</p> <p>Default: ACTR.</p>						
<i>error</i>	<p>Transient data queue name where error messages information is logged.</p> <p>Default: ACER.</p>						
RETRYINT	<p>Interval between restart attempts in seconds.</p> <p>Acceptable values are five through 86400.</p> <p>Default: 120 (Two minutes).</p>						
RETRYMAX	<p>Number of times Socket Management attempts to restart if the TCP provider is not available.</p> <p>Acceptable values are 0 through 999. A value of zero means that no restart is attempted.</p> <p>Default: Zero.</p>						

SCTYEXIT	The name of the security exit program to execute when a new connection is created by any listener. See the appendix “Security Program” for more information.
SCTYTYPE	How the security exit program that gets control if SCTYEXIT is coded on the T09MLSTM and T09MCICS macros. <i>normal</i> The security program name specified in the T09MLSTM macro SCTYEXIT parameter can override what is coded on the T09MCICS SCTYEXIT parameter. <i>mandatory</i> The security exit program name specified in the SCTYEXIT parameter of the T09MCICS macro – it is always used. Note: Anything specified at the listener T09MLSTN macro will be ignored. See the appendix “Security Program” for more information. Default: Normal.
SCKTVIEW	Required. Yes Enables (starts) Socket Management. No Disables Socket Management. Default: YES
SSN	Unicenter TCPaccess only. Subsystem identifier for the Socket Management routines to use in the TCP subsystem identifier. <i>ssn</i> defines the subsystem identifier of the Unicenter TCPaccess address space – this is the SSN parameter on the execution JCL of Unicenter TCPaccess. Default: ACSS.
TRANSID	Defines transaction IDs used in Socket Management operation. <i>stop</i> Transaction ID for the program T09TTERM as defined to the CICS/TS RDO definitions. This transaction ID disables the currently active Socket Management environment. Default: IPPR.

<i>lstn</i>	<p>Transaction ID for the program T09TLSTN as defined to the CICS/TS RDO definitions. It must be specified if any T09MLSTN macros follow in the configuration. This transaction starts a Socket Management Listen tool.</p> <p>Default: IPTL.</p>
<i>strt</i>	<p>Transaction ID for the program T09TSTRT as defined to the CICS/TS RDO definitions. This transaction ID enables the Socket Management environment; starts listeners; and starts the NetMaster interface.</p> <p>Default: IPST.</p>
<i>lst2</i>	<p>Transaction ID for the program T09TLST2 as defined to the CICS/TS RDO definitions. This transaction is the second half of the <i>lstn</i> transaction and is used when the security or Client-Data option is in effect.</p>
TRANSTBL	<p>Default translation table name for all of the following configuration statements. This load module must be defined to CICS/TS RDO, or Socket Management initialization will fail.</p>
<i>tblname</i>	<p>Specifies the name of the default ASCII to EBCDIC and EBCDIC to ASCII translation table for the rest of the configuration. This can be any one of the translate tables provided with the product.</p> <p>Default: T09XENG.</p>
TRCSSN	<p>Subsystem identifier for the trace address space subsystem identifier. When the trace address space has been started one can use TCPEEP to trace EZASOKET or EZACICAL calls and their results.</p> <p>Default: ACTR.</p>
USERID	<p>User ID that the LISTEN API or LISTEN tool uses when starting child transactions. This allows these child transactions to inherit the security permissions of the specified user ID.</p> <p>When this parameter is specified on the T09MCICS statement, it takes effect for all LISTEN API or LISTEN tools that do not specify a USERID parameter.</p>
USRTRNID	<p>Transaction ID for the program T09TULST. It is used to start a particular T09MTRAN entry in the T09CONxx configuration file.</p> <p>Default: IPUL.</p>

Changing Translation Tables

The translation table specified in the T09MCICS configuration macro is used for all translation functions.

To use an alternate translation table do the following:

1. Create a valid program table entry in an installed Resource Definition Online (RDO) group.
2. Concatenate the load module in which the table resides into DFHRPL for the CICS/TS region.
3. Perform the steps in the “Customization” chapter on customizing the translation table.

T09MLSTN

The T09MLSTN macro defines a port for a TCP listener in CICS/TS. When a connection is made, a defined data processing (child) transaction is initiated. Optional buffering specifications, statistic, and tracing options can be defined.

Note: There is no limit to the number of T09MLSTN canned listener macros that you can configure and start. However, for each macro coded, you will generate one long running transaction in CICS/TS.

IBM TCP/IP only: For every T09MLSTN coded in the T09CONxx configuration file you may need to define port security in the PORT section of the profile.tcpip data set as follows:

```
1234 TCP cicsprod
```

Where:

1234 The PORT=1234 parameter.

cicsprod The name of your CICS/TS started task.

```
T09MLSTN
  [ CLNTIME = seconds ]
  [ , CLNTLEN = data_len ]
  [ , CLNTRNS = YES | NO ]
  [ , CLNTTBL = translation_table ]
  [ , MSOCK = max_sockets ]
  [ , PORT = ( number ) ]
  [ , QLSTN = num_backlog_queue ]
  [ , SCTYEXIT = exit-program-name ]
  [ , USERID = userid ]
```

CLNTIME	<p>Client-Data Listener option. Specifies that the listen call should receive the input data stream to determine which transaction ID to start. Refer to Client/Data Listener Option section for the required input formats.</p> <p>The CLNTIME value indicates the number of seconds the LISTEN tool waits to receive the input parameter data stream from the client, once the client successfully establishes a connection with the LISTEN tool.</p> <p>Default: Zero (No Client-Data Listener option).</p>
CLNTLEN	<p>Maximum length of data the LISTEN tool tries to receive for the initial data stream.</p> <p>This value is useful when the amount of client data being sent is known and consistent. See Client/Data Listener Option for the required input formats.</p> <p>Default: Initial data stream is a maximum length of 50 bytes.</p> <p>This greatly speeds processing by enabling the LISTEN tool to continue processing new connections without waiting the full CLNTIME value for the initial data.</p> <p>Note: This length includes the entire data stream from the beginning of the transaction name through the actual client data bytes including any imbedded commas.</p>
CLNTRNS	<p>CLNTRNS=YES indicates that the initial input stream should be translated.</p> <p>Default: CLNTRNS=NO.</p>
CLNTTBL	<p>Name of the translation table to use for translating the input stream. If no translation table is specified, the default translation table is used.</p>
MSOCK	<p>Maximum number of concurrent sockets per initapi socket calls the product supports.</p> <p>Typically, each connection requires a socket. In general, every server and every individual transaction calls initapi or has one implied by doing a socket function that requires an initapi.</p> <p>Note: Only one initapi call is implied per CICS/TS transaction number.</p> <p>Maximum: 2000.</p> <p>Default: 50.</p>

PORT Transport provider port number to listen on for connections. number defines the transport provider port number to listen on. This field or the service name field is required.

Note: PORT and SERVICE are mutually exclusive.

QLSTN Listen Backlog Queue.

Default: Five.

SCTYEXIT Security exit program name to use when a connection is processed by this listener. It may not be executed if the T09MCICS macro has SCTYTYPE=MANDTORY coded.

See the appendix "Security Program" for more information.

Client/Data Listener Option

The Socket Management Client/Data Listener option allows one listening TCP/IP socket port to serve as a multi-function server. This is achieved by passing the CICS/TS transaction name in the initial TCP packet. In this way a single server can distribute connections to many different applications.

This server is compatible with applications written to use IBM's CICS/TS provided listener CSKL.

It should be noted that this additional server flexibility does have a performance impact. By having the listener do a receive as part of it's processing, the servicing of new connections can be delayed. For this reason, this listen server type is not recommended for high connection volume services.

In an attempt to avoid many of the inherent performance problems, the client/data listener tool service is broken into two transactions:

- The first transaction handles connection establishment thus blocking the port for a minimal amount of time.
- The second phase of the listener, waits for the client data independent of blocking the connection establishment port.

To further enhance performance you should consider using the CLNTLEN parameter whenever possible. Another way to improve performance is to start multiple client/data listeners. That way any high volume applications can be on their own server port independent of low volume applications. There are no restrictions to the number of client data listeners that can be started. In following these suggestions, any possible performance issue can easily be eliminated.

The design of the Client/Data Listener mimics the format of a standard CICS/TS 3270 terminal data stream. In other words, this is very similar to what you are used to seeing come into a standard CICS/TS terminal interaction on initialization of a terminal transaction. The first four characters of the initial data packet is the transaction name as if you were coming from a real 3270 CICS/TS terminal. Another similarity is that the transaction name can be followed by optional data (parameters) that are passed to the transaction. This is a great listener to have for providing multiple applications with TCP connectivity within just one long running server transaction. Refer to the previous performance notes for other considerations.

The client data listener works in the following manner:

When a connection is received, the phase two listener is started to free up (unblock) the original server listening port.

The phase two listener:

- Does a TCP receive from the network
- Expects one of the following client data formats to be received:

```
TRAN
TRAN, UUUUUUUUUUUU
TRAN, UUUUUUUUUUUU, IC, HHMMSS
TDQN, UUUUUUUUUUUU, TD
TRAN, , IC, HHMMSS
TDQN, , TD
```

Depending on the format of data, the listener determines how the actual spawned application child is started. Continue reading for further details on how this works.

The client data listener is turned on by coding a value in the CLNTTIME field that is greater than zero. There are also options for translating the client data string and what translation table if that is desired.

Default: No translation.

TRAN | TDQN Can be a one- to four-character field followed by an optional comma implying more parameters. The field can contain one of the following:

- A transaction ID that is to be started
- A transient data queue (TDQ) name to which the 1- to 35-bytes of optional user data will be written if provided

UUUUUUUUUUUUUU

A 1- to 35-bytes of user data to be passed to the started transaction or written to the transient data queue in the field CLNTDATA.

IC Specifies that transaction TRAN be started in HHMMSS.

Note: If left blank, start up is immediate.

HHMMSS Hours, minutes, and seconds for the IC option.

TD Indicates that the optional client data field CLNTDATA(UUUUUUUUU above) will be written into the transient data queue, TDQN.

Client/Data Option Data Structure

The data structure passed to the invoked program has the following format. This structure is accessed via performing a EXEC CICS RETREIVE command in the invoked(spawned child) transaction.

```

CLNTPARM DS 0F
TOKEN DS F New token - socket ID
DS DS CL16 Reserved
CLNTDATA DS CL36 Up to 35 bytes of client data
PROTADDR DS 0F
DOMAIN DS H Family
RPORT DS H Remote port
RADDR DS F Remote IPADDR
DS DS XL8'00' Reserved
CLNTLEN EQU *-CLNTPARM
    
```

Examples

Client/Data Listener with Translation

To invoke the Client/Data Listen Tool and automatically translate the input stream from ASCII to EBCDIC, you must specify the following options in the T09MLSTN parameter:

```
T09MLSTN PORT=2002,CLNTIME=5,CLNTRNS=YES,CLNTTBL=MYTABLE
```

In this example, the Listen tool:

- Listens for connections on port 2002
- Waits for up to five seconds for the input stream after establishing a connection
- Translates the input stream using the translation table MYTABLE

T09MCMDS

The T09MCMDS macro defines the interface from Unicenter NetMaster to Socket Management.

Note: Coding this macro to enable the interface will generate one long running transaction in CICS/TS.

IBM TCP/IP only: You will need to define port security in the PORT section of the profile.tcpip data set as follows:

```
1234 TCP cicsprod
```

Where:

1234 The PORT=1234 parameter.

cicsprod The name of your CICS/TS started task.

```
T09MCMDS PORT=port,                      X ]
[ , TRANSID=transid,                      X ]
[ , SECURITY=secopt,                      X ]
[ , SECNAME=secentity,                   X ]
[ , LOG=logopt,                          X ]
[ , MSOCK=maxsock,                      X ]
[ , QLSTN=qlstn,                         X ]
[ , TERMID=termid                        ]
```

PORT=*port* Required. The port on which the NetMaster command processor listens.

Default: None.

TRANSID=*transid* The NetMaster command processor transaction ID. This parameter must match the definition for program T09TCMDS in the CICS/TS RDO definitions. This is the CICS/TS transaction used during Socket Management initialization to launch the command server.

Default: IPCP.

SECURITY=Y | N SECURITY=Y specifies whether the command server will validate a user ID and password combination for all users. Should the user provide a valid user ID/password combination then the user's authority to the SECNAME entity is validated.

The terminal configured in the TERMID parameter must be available for use in the CICS/TS region whenever SECURITY=Y has been set.

SECURITY=N allows any user of the command server to utilize all commands.

***Important!** You may need to disable security in a testing environment, but it is strongly recommended that you always enable security in a production environment.*

Default: Y.

SECNAME=secentity The name of the external security system (ESM) general resource entity the command processor uses to verify user command authority. The command processor issues an ESM/SAF call to verify users have READ or UPDATE access against this resource.

The name of this resource can be up to 44 characters and must be filed in the ESM's general resource profile (for example, FACILITY).

- If users have READ access, they are allowed to display session, server and global statistics
- If users have update access, in addition to displaying session, server and global statistics, they are also allowed to start and stop transactions, sessions, servers and applications

Note: If omitted, the default of \$SKTVIEW.CICS.COMDAUTH is used.

LOG=logopt Specifies whether non-critical command server events are recorded in the log.

Specify one or more of the following keywords within parentheses separated by commas:

SEC Record security violation messages in the log.

INFO Record informational messages in the log.

ERROR Record non-critical error messages in the log.

Note: Critical error messages are always recorded in the log regardless of *logopt*. To record **all** events, specify LOG=(SEC,INFO,ERROR).

Default: ERROR.

MSOCK=*maxsock* Specifies the maximum number of concurrent administrative connections supported by the command server. Specify a number between 50 and 2000. The

Default: 50.

QLSTN=*qlstn* Specifies the listen queue depth (backlog).

This is the maximum number of concurrent connection requests permitted in the listen request queue. When this queue is full, subsequent connection requests are discarded.

Default: Five.

TERMID=*termid* Specifies the name of the terminal with which to associate the command server. See Defining the NetMaster Command Processor's Principal Facility in the "Configuration" chapter for more information.

Default: TCMD.

T09MTRAN

The T09MTRAN macro defines a CICS transaction that can be started by Socket Management or CPT. It is an excellent mechanism to start non T09MLSTN servers during CICS startup.

Note: There is limit of 255 T09MTRAN entries in the T09CON:xx configuration file.

T09MTRAN macro and its parameters:

```
T09MTRAN
[ TRANSID = transaction to start ]
[, USERID = userid ]
[, TERMID = CICS terminal ]
[, IMMED = YES | NO ]
[, ID = unique 1-8 character ID ]
[, APPLID = CICS VTAM APPLID ]
[, PORT = ( number ) ]
[, BACKLOG = num_backlog_queue ]
[, ACCTIME = accept() timeout ]
[, REATIME = read() timeout ]
[, GIVTIME = givesocket() timeout ]
[, NUMSOCK = num_sockets ]
[, MINMSGL = minimum message length ]
[, TRANTRN = YES | NO ]
[, TRANUSR = YES | NO ]
[, SCTYEXIT = exit-program-name ]
[, WLMGN1 = WLM Group name 1 ]
[, WLMGN2 = WLM Group name 2 ]
[, WLMGN3 = WLM Group name 3 ]
```

TRANSID	<p>Transaction ID to start. This is a mandatory parameter.</p> <p>Default: None.</p>
USERID	<p>User ID to be used when starting the transaction. This allows the transaction to inherit the security permissions of the specified user ID.</p> <p>Note: The USERID parameter cannot be specified with the TERMID parameter.</p> <p>Default: None.</p>
TERMID	<p>Specifies the name of the terminal to associate with the started transaction.</p> <p>The TERMID parameter cannot be specified with the USERID parameter.</p> <p>Default: None.</p>
IMMED= YES NO	<p>States whether the transaction should be started at product startup.</p> <p>Default: YES.</p>
ID	<p>Unique one- to eight-characters ID used to uniquely identify an entry. If you allow T09MTRAN to default, it will generate unique character IDs of the form ID appended with the instance of the T09MTRAN macro in the T09CONxx configuration file.</p> <p>Default: "ID" appended with the instance of the T09MTRAN macro in the T09CONxx configuration file.</p>
APPLID	<p>Specifies the name of the CICS VTAM APPLID to run the transaction in at product startup. If you let the APPLID default, it will run at product startup (depending on the IMMED specification). If you place a parameter on the APPLID field then the APPLID must match the CICS VTAM APPLID in the running CICS region before the transaction can be started in the region at product startup.</p> <p>Default: Eight blanks.</p>
PORT	<p>Port Number found in the CFLPORT field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the PORT parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values: 1-65535.</p> <p>Default: None.</p>

BACKLOG	<p>Listen Backlog Queue to be found in the CFLBKLOG field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the BACKLOG parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values: 0-999.</p> <p>Default: 20.</p>
ACCTIME	<p>accept timeout in seconds, found in the CFLLTME field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the ACCTIME parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values: 0-999.</p> <p>Default: 60.</p>
REETIME	<p>read timeout in seconds, found in the CFLRTME field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the ACCTIME parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values: 0-32767.</p> <p>Default: Zero.</p>
GIVTIME	<p>givesocket() timeout in seconds, found in the CFLGTME field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the GIVTIME parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values: 0-999.</p> <p>Default: 60.</p>

NUMSOCK	<p>Specifies the maximum number of concurrent connections supported, found in the CFLNSOCK field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the NUMSOCK parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values: 50-2000.</p> <p>Default: 50.</p>
MINMSGL	<p>Specifies the minimum input message length, found in the CFLNMIN field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the MINMSGL parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Valid values are 4-99.</p> <p>Default: Four.</p>
TRANTRN= YES NO	<p>States whether the transaction ID should be translated, found in the CFLOPTTR field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the TRANTRN parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction</p> <p>Default: None when no other CFG0000 parameters are specified. The default is YES when other CFG0000 parameters are specified.</p>
TRANUSR= YES NO	<p>States whether user data should be translated, found in the CFLOPTUD field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the TRANUSR parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction</p> <p>Default: None when no other CFG0000 parameters are specified. The default is YES when other CFG0000 parameters are specified.</p>
SECEXIT	<p>Security exit program name, found in the CFLSECEX field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the MINMSGL parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Default: Eight blanks.</p>

WLMGN1	<p>WLM Group name 1, found in the CFLWLMN1 field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the WLMGN1 parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Default: 12 blanks.</p>
WLMGN2	<p>WLM Group name 2, found in the CFLWLMN2 field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the WLMGN2 parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Default: 12 blanks.</p>
WLMGN3	<p>WLM Group name 3, found in the CFLWLMN3 field addressed by the CFG0000 DSECT from the LSTCFGDA pointer.</p> <p>Note: Specifying the WLMGN3 parameter forces the LSTP parameter DSECT to be passed to the TRANSID transaction.</p> <p>Default: 12 blanks.</p>

Parameter List Passed to T09MTRAN Transactions

There are two possible layouts of data passed to the customer's transaction:

- Simple data from the PARM field.
- The LSTP DSECT with a length of 32 bytes.

Regardless of the parameter list the application issues a CICS retrieve call to access the data:

```
EXEC CICS RETRIEVE SET() LENGTH()
```

The first and recommended default layout is to only pass the information from the PARM= field in the T09MTRAN macro. This layout applies anytime that a user listener application has not configured any of the CFG0000 fields in the T09MTRAN macro. In this case, the only fields you can code are the USERID, TERMID, TRANSID, ID, APPLID and PARM in the T09MTRAN macro.

The LSTP parameter DSECT is generated when ever any of the CFG0000-related parameters are specified on the T09MTRAN macro:

PORT	GIVTIME	TRANUSR	WLMGN3
BACKLOG	NUMSOCK	SECEXIT	
ACCTIME	MINMSGL	WLMGN1	
REACTIME	TRANTRN	WLMGN2	

LSTID	DS	CL4'LSTP'	ID for Control block
LSTTLEN	DS	H	Total length of LSTPARMS DSECT
LSTVERS	DS	XL2	Version number
LSTOPTDA	DS	A	pointer to the optional Data
LSTOPLN	DS	F	Length of the optional Data
LSTCFGDA	DS	A	pointer to the CFG0000 field
LSTCFLN	DS	F	Length of the CFG0000 field
LSTLCADA	DS	A	pointer to the current LCA
LSTLCALN	DS	F	Length of the current LCA

T09MEND

The T09MEND macro terminates lists of configuration parameters.

Important! This macro has no parameters, but is required and must be the last macro statement in the configuration.

CICS/TS Resource Definition Reference

This chapter provides information for the installation of the required CICS/TS Resource Definitions (RDO) needed for Unicenter NetMaster Socket Management for CICS (Socket Management).

It includes these sections:

- [System Initialization Tasks \(SIT\) Entries](#)
- [Program List Table \(PLT\) Entries](#)
- [Sample CICS/TS RDO Member T09RDO](#)
- [Transient Data Queue \(TDQ\) Entries](#)
- [Program Entries](#)
- [Terminal Control \(TCT\) Entries](#)

Note: See *Getting Started* for software installation instructions.

Socket Management definition statements must be added to the CICS/TS Resource Definition Online (RDO) and to resource definition tables. Program, transaction ID, and destination definition statements are required. Some destination and PLT definition statements are recommended, but are optional.

You must define the Socket Management program names for system initialization, termination, Task-Related User Exits (TRUEs), and configuration to CICS/TS RDO. These programs have required transaction IDs that must be defined. The transaction IDs are distributed with default values, but can be modified during Socket Management installation or customization.

A sample DFHCSDUP input file, member name T09RDO, is provided in the T09SAMP library for defining program, transaction, and destination entries using CICS/TS RDO.

It is recommended that you define the initialization program in the PLTPI table. Alternatively, the initialization program can be started interactively by its transaction ID (see the chapter “Operations”). The termination program *must* be defined in the PLTSD table. The termination program provides proper release of transport provider connections and Socket Management resources during CICS/TS shutdown. The Socket Management termination and initialization program transaction IDs can be used interactively at a CICS/TS terminal to control Socket Management.

Socket Management uses transient data queues (TDQ) for logging support and for the Socket Management tools facility. Socket Management logs informational errors, statistics, and traces messages to transient data queues. These TDQs can be existing or new queues. If the TDQs are new, they must be defined as shown in sample T09RDO entries.

Socket Management definition statements must be in the appropriate tables either by configuring them, or by making them dynamically available through RDO commands. You may need to restart CICS depending on the installation method you used.

System Initialization Tasks (SIT) Entries

During Socket Management installation or customization, pay close attention to the maximum active tasks (AMXT) and maximum number of concurrent tasks (MXT) values. The Socket Management server application, the listen tool, is designed to be a long-running CICS transaction. Additionally, a server can be written to start any number of concurrent data processing tasks, the automated listening transaction is such a task. Therefore, installations should monitor active Socket Management applications for the number of concurrent, active tasks.

The default product configuration table is T09CONFIG. A site that runs the product with a non-default T09CONxx configuration file may pass the two-digit suffix in the CICS/TS SIT table INITPARM parameter overrides. For example, place the following INITPARM entry in the SIT overrides to start the product using the T09CONA2 configuration table during CICS startup:

```
INITPARM=(T09TSTR='A2')
```

Loads configuration table T09CONA2.

Program List Table (PLT) Entries

The initialization routine resource definition shown below can be included in the PLTPI. Depending on the release, the Socket Management initialization routine must appear after the entry statement for DFHDELIM or during the third stage of initialization. The resource definition statement for the T09TSTRT program is optional, but is generally recommended.

A default PLTPI table entry can be copied from the T09SAMP library member T09PLTPI.

Note: Refer to Configuration Table Suffixing in the “Customization” chapter for details on alternate configuration table names.

```

*-----*
*
*      Socket Management & CPT  PLT (PLTPI=) ENTRY      *
*
*      THESE PLT ENTRIES ARE FOR THE SYSTEM INITIALIZATION PLT *
*      TABLE.                                           *
*-----*
*
*      STARTUP ENTRY                                     *
*
*      DFHDELIM                                         *
*
*      DFHPLT TYPE=ENTRY,          Socket Management/CPT START UP
*      PROGRAM=T09TSTRT
*

```

This termination routine resource definition must be included in the shutdown section of the PLTSD. Depending on the release of CICS/TS, the Socket Management termination routine must appear **before** the entry statement for DFHDELIM or during the first stage of termination. This program provides proper termination of the Socket Management Interface during CICS/TS shutdown. This includes the release of all transport provider connections and disabling the Socket Management Interface.

A default PLTSD table entry can be copied from the T09SAMP library member T09PLTSD.

```

*-----*
*
*      Socket Management / CPT PLT (PLTSD=) ENTRY      *
*
*      THIS PLT ENTRY IS FOR THE SYSTEM TERMINATION PLT TABLE. *
*-----*
*
*      DFHPLT TYPE=ENTRY,          X
*      PROGRAM=T09TTERM
*

```

Note: If you create PLTPI and PLTSD entries from these samples only, you must add TYPE=INITIAL and TYPE=FINAL statements.

Sample CICS/TS RDO Member T09RDO

The default CICS/TS RDO entries can be copied from the T09SAMP library member T09RDO, which is shown below. Installing these definitions is required for Socket Management. As noted in the comments at the top of the T09RDO member a "?" was inserted in front of the delete group command to prevent accidental deletion. It is recommended that Unicenter NetMaster Socket Management for CICS have its own RDO group, so that CICS/TS RDO entries can be deleted and refreshed with any new releases of the product.

Note: If you follow this procedure, blank out the "?", and install the group below. This group must be part of the RDO list that starts up this CICS/TS region.

```

*-----*
*
*   Socket Management and CICS PROGRAMMER'S TOOLKIT RDO ENTRIES
*
*-----*
*
* The below "?" is intended to cause a failure so that this sample
* does not accidentally delete an existing RDO group.  If you desire
* to start with fresh definitions which are needed with this new
* version you will need to blank out the "?".  Make sure first,
* that any extra definitions within this group are saved within
* another group if they are needed.  If you simply want to add the
* new entries to the existing group then delete the following line and
* submit this member as is, this will cause existing entries to fail
* for dups and the new entries to be added.
*
? DELETE GROUP (T09CPT)
*
*-----*
*
*   Socket Management and CICS PROGRAMMER'S TOOLKIT PROGRAMS
*
*-----*
*
DEFINE PROG (T09COMON) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09CONFG) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09CONCP) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09CONEZ) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TSTR) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TTERM) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TCMDS) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TLOID) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TLSTN) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TLST2) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TMROS) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TREC) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TSEND) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TSLCT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09XENG) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TIPCK) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09IUEXT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ATADD) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ABDTL) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)

```

```

DEFINE PROG (T09ACNFG) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ADMGR) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AEXIT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AGENT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AOLWT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AHELP) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ATLST) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AMAIN) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AMENU) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AOCTL) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09APING) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AQCLS) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ARSTQ) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ATSND) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09ATROP) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09AUTIL) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AWTCH) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AYANK) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09AZAPS) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FCLOS) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FCONN) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FGIVE) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FLFTP) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FLSTN) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FRCFR) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FRECV) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FSEND) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FSLCT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FSNTO) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FTAKE) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09FXLAT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09MAPS) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09MAPT) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09MAPU) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09NMEVX) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TASC1) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TASC2) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
DEFINE PROG (T09TCFCM) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TCFDG) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TCFDM) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (T09TCFRM) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
EXECK (CICS)
DEFINE PROG (EZACIC01) LANG (ASSEMBLER) DA (ANY) GROUP (T09CPT)
*
*-----*
*
* Socket Management and CICS PROGRAMMER'S TOOLKIT TRANSACTIONS *
*-----*
*
DEFINE TRANS (IPST) PROG (T09TSTRT) TASKDATALOC (ANY) GROUP (T09CPT)
TASKDATAK (CICS)
DEFINE TRANS (IPPR) PROG (T09TTERM) TASKDATALOC (ANY) GROUP (T09CPT)
TASKDATAK (CICS)
DEFINE TRANS (IPTL) PROG (T09TLSTN) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPT2) PROG (T09TLST2) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)
DEFINE TRANS (IPTR) PROG (T09TREC) TASKDATALOC (ANY) GROUP (T09CPT)

```



```

* SEND (CLIENT) OPTION=ALL *
DEFINE TDQUEUE(IPSA) TYPE(INTRA) TRIGGERLEVEL(1) TRANSID(IPTS)
      RECOVSTATUS(LOGICAL) ATIFACILITY(FILE) GROUP(T09CPT)
* SEND (CLIENT) OPTION=SEP (CR/FL) *
DEFINE TDQUEUE(IPSS) TYPE(INTRA) TRIGGERLEVEL(1) TRANSID(IPTS)
      RECOVSTATUS(LOGICAL) ATIFACILITY(FILE) GROUP(T09CPT)
*
*-----*
* LIST OBJECTS *
*-----*
*
LIST GROUP(T09CPT) OBJECTS

```

Transient Data Queue (TDQ) Entries

The required TDQ entries are installed as described previously in section Sample CICS/TS RDO member T09RDO.

Unicenter NetMaster Socket Management for CICS uses transient data queues for logging support information. The log transient data queues are defined in the T09MCICS macro instruction.

The transient queues are configured with default names. These names can be modified, but the corresponding name must be changed in the default configuration file T09CONez in macro T09MCICS.

Note: The Socket Management queues by default are redirected (indirect) to CSML queue, which is often redirected (indirect) to CSSL, which in turn is frequently redirected to SYSOUT. The redirection is expected and works fine. The above is how the sample definitions are shown. However, if a data set is the final destination allocated, then the data set should have variable length records of at least 121 bytes.

Program Entries

The required program entries are installed as previously described in Sample CICS/TS RDO member T09RDO.

Processing program statements must be defined for all Socket Management base product programs.

These programs are:

- System initialization
- Termination
- TRUE routines
- Configuration
- Translation table
- Support routines

All Socket Management base product programs are written in assembler.

Note: Unicenter NetMaster Socket Management for CICS base product program names cannot be modified.

Use RDO and the information in the following table of Socket Management base product program definitions to define base product programs.

Program	Language	Description
T09COMON	Assembler	TRUE routines.
T09CONCP	Assembler	Configuration table for CPT.
T09CONEZ	Assembler	Configuration table for Socket Management.
T09TCMDS	Assembler	NetMaster command processor transaction (must run with EXECKey = CICS).
T09TSTRT	Assembler	Initialization transaction (must run with EXECKey = CICS).
T09TTERM	Assembler	Termination transaction.
T09TLSTN	Assembler	Automated listening transaction.
T09TLST2	Assembler	Automated listening transaction phase 2.
T09TMROS	Assembler	MRO program.
T09TREC V	Assembler	Automated receive transaction.

Program	Language	Description
T09TSEND	Assembler	Automated send transaction.
T09TSLCT	Assembler	Select tool transaction.
T09TLOID	Assembler	Initialization processing transaction (must run with EXECKey = CICS).
T09TIPCK	Assembler	IVP transaction.
T09XENG	Assembler	Translation table.
T09IUEXT	Assembler	Optional exit program.
T09ATADD	Assembler	Add/Alter tool.
T09ABDTL	Assembler	Browse detail – clients or servers.
T09ACNFG	Assembler	Configuration summary.
T09ADMGR	Assembler	Screen manager.
T09AEXIT	Assembler	Exit interface.
T09AGENT	Assembler	Generate trace.
T09AOLWT	Assembler	Online queue write.
T09AHELP	Assembler	Help presentation.
T09ATLST	Assembler	Listen tool display.
T09AMAIN	Assembler	Main entry manager.
T09AMENU	Assembler	Main menu display.
T09AOCTL	Assembler	Operations menu.
T09APING	Assembler	Ping remote host.
T09ARSTQ	Assembler	Reset trace/error queues.
T09ATSND	Assembler	Send tool display.
T09ATROP	Assembler	Add/alter trace options (must run with EXECKey = CICS).
T09AUTIL	Assembler	Utilization summary.
T09AWTCH	Assembler	Online queue display.
T09AYANK	Assembler	Online queue release.
T09AZAPS	Assembler	Reset statistics.

Program	Language	Description
T09MAPS	Assembler	Basic mapping support screens.
T09MAPT	Assembler	Basic mapping support screens.
T09MAPU	Assembler	Assembler.

Transaction Entries

The required transaction entries are installed as previously described in Sample CICS/TS RDO member T09RDO.

Program control statements must be defined for Socket Management base product transactions. These transactions consist of system initialization, termination, automated tool routines and support routines.

The Unicenter NetMaster Socket Management for CICS base product is distributed with default transaction IDs. These transaction IDs are specified within the Socket Management configuration file T09CONez. If the default transaction IDs are modified then the change must be reflected in the configuration file.

Use RDO and the information in the following table of Socket Management base product transaction ID definitions to define Socket Management base product transaction IDs.

Program	Trans ID	Description
T09TSTRT	IPST	Initialization transaction. Only these transactions can be issued directly by a user. Must run with TASKDATAKey = CICS
T09TTERM	IPPR	Termination transaction. Only these transactions can be issued directly by a user.
T09TSLCT	IPSL	Select Tool transaction.
T09TLSTN	IPTL	Automated listening transaction.
T09TLST2	IPT2	Automated listening transaction phase 2.
T09TRECV	IPTR	Automated receive transaction.
T09TSEND	IPTS	Automated send transaction.
T09TIPCK	IPCK	IVP transaction.

Program	Trans ID	Description
		Only these transactions can be issued directly by a user.
T09TIPCP	IPCP	NetMaster command processor transaction.
T09ATADD	IPAA	Add/alter tool.
T09ABDTL	IPAB	Browse detail – clients or servers.
T09ACNFG	IPAC	Configuration summary. Only these transactions can be issued directly by a user.
T09AQCLS	IPAE	Close connections.
T09AGENT	IPAG	Generate trace.
T09AOLWT	IPAI	Online queue write.
T09AHELP	IPAK	Help presentation.
T09ATLST	IPAL	Listen tool display.
T09AMAIN	IPAM	Main entry manager. Only these transactions can be issued directly by a user.
T09AMENU	IPAN	Main menu misplay.
T09AOCTL	IPAO	Operations menu. Only these transactions can be issued directly by a user.
T09APING	IPAP	Ping remote host.
T09AQCLS	IPAQ	Close connections.
T09ARSTQ	IPAR	Reset trace/error queues.
T09ATSND	IPAS	Send tool display.
T09ATROP	IPAT	Add/alter trace options.
T09AUTIL	IPAU	Utilization summary. Only these transactions can be issued directly by a user.
T09AWTCH	IPAW	Online queue display.
T09AYANK	IPAY	Online queue release.
T09AZAPS	IPAZ	Reset statistics.

Terminal Control (TCT) Entries

You must define a principal facility to be associated with the NetMaster command processor interface.

The NetMaster command processor interface requires a principal facility, a terminal, to be associated with it in order for its security checking to work properly. You can define the terminal via the standard CICS/TS terminal definition macro, DFHTCT. The NetMaster command processor never directly references this dummy terminal and it is not used for traditional terminal purposes.

- The terminal name and terminal type you select must be valid as far as CICS/TS is concerned, however they are meaningless to The NetMaster command processor.
- The NetMaster command processor is configured in T09CONez macro T09MCMDS, see the “Configuration Statement Reference” chapter for more information.
- In the T09MCMDS macro you can select a terminal name via the TERMID= parameter or let it default to TCMD. The TERMID= *parameter* of the T09MCMDS macro must match the TRMIDNT= *value* in the DFHTCT entry defining the terminal.

Note: In the sample below this field is shown in *bold italics*.

- You must also include a dummy DD statement for this terminal in your CICS/TS region startup JCL.

```
// PRNT001 DD DUMMY
```

The default CICS/TS TCT entries can be copied from the T09SAMP library member T09TCT, which is shown below.

```
TCTJH  TITLE 'DFHTCT MASTER TCT'
        PRINT GEN
        DFHTCT TYPE=INITIAL,                                X
                ACCMETH=(NONVTAM,VTAM),                    X
                SUFFIX=JH
        DFHTCT TYPE=SDSCI,DEVICE=1403,DSCNAME=PRNT001
        DFHTCT TYPE=LINE,ACCMETH=BSAM,INAREAL=80,TRMTYPE=CRLP, X
                OSADSCN=PRNT001
        DFHTCT TYPE=TERMINAL,TRMIDNT=TCMD,ERRATT=NO,LPLEN=80, X
                PGESIZE=(24,80),TRMSTAT=RECEIVE
        DFHTCT TYPE=FINAL
        END  DFHTCTBA
```

Unicenter NetMaster Command Processor Security

Unicenter NetMaster command processor server security is activated via T09MCMDS configuration and, by default, is fully enabled. The T09MCMDS configuration is described in the chapter “Configuration Reference.” Security is optional, however it is highly recommended that security **not** be disabled to ensure that unauthorized users are denied access.

Unicenter NetMaster command processor server security ensures that users have valid user IDs and passwords, and that users have the correct authority level to enter command server requests.

When the command processor security is enabled (by specifying SECURITY=Y on the T09MCMDS macro in the T09CONxx configuration file), all users requesting a connection with the command server must have:

- A valid user ID and password registered with the ESM (external security manager) under which the Unicenter NetMaster command processor is running.
- An entry in the ESM's general resource class profile (for example, FACILITY) must be defined so the command processor can verify that even valid users have the proper authority to enter command server requests.

The facility resource name can be up to 44 characters and must be configured in the ESM's general resource profile (FACILITY) and on the SECENT parameter of the T09MCMDS macro in the T09CONxx configuration file.

There are two levels of FACILITY command authority access:

- READ access: Allows users to issue commands to display session, server and global statistics.
- UPDATE access: Allows users to issue commands to alter (and display) the CICS environment, and:
 - Stop and start servers
 - Terminate sessions
 - Shutdown and restart the product
 - Initiate CICS transactions

Security Configuration for eTrust CA-ACF2

Many sites have both production and test CICS regions. You should give operators and system personnel the proper authority to do their job. At the same time, you want to protect the CICS applications from overly inquisitive or unruly personnel in the user community.

The 44 characters resource name must match the SECENT parameter of the T09MCMDS macro in the T09CONxx configuration file. Many sites have a requirement that personnel have authority levels appropriate for both production and development CICS regions.

Once you define these entities in the FACILITY resource class, you must then give personnel the proper authority to do their job.

The default access of NONE prevents users from using the Unicenter NetMaster command/control server.

READ access allows users to display session, server and global statistics.

UPDATE access enables users to start and stop transactions, sessions, servers and applications.

The following are sample FACILITY entity definitions for use by MVS system personnel (SYS1), where:

CICSOPR A CICS operator

CICSDEV A CICS developer

\$SKTVIEW.CICSPROD.COMDAUTH

 Configured in the SECENT parameter for the CICS production regions

\$SKTVIEW.CICSTEST.COMDAUTH

 Configured in the SECENT parameter for the CICS development regions

eTrust CA-ACF2
Example

```
SET R(FAC)
COMP
$KEY($SKTVIEW.CICSPROD.COMDAUTH) TYPE(FAC)
UID(SYS1-)                    SERVICE(READ,UPDATE)
UID(CICSOPR-)                SERVICE(READ,UPDATE)
UID(CICSDEV-)                SERVICE(READ)
STORE
COMP
$KEY($SKTVIEW.CICSTEST.COMDAUTH) TYPE(FAC)
UID(SYS1-)                    SERVICE(READ,UPDATE)
UID(CICSOPR-)                SERVICE(READ,UPDATE)
UID(CICSDEV-)                SERVICE(READ,UPDATE)
STORE
```

Security Configuration for eTrust CA-Top Secret

Many sites have both production and test CICS regions. You should give operators and system personnel the proper authority to do their job. At the same time, you want to protect the CICS applications from overly inquisitive or unruly personnel in the user community.

The 44 characters resource name must match the SECENT parameter of the T09MCMDS macro in the T09CONxx configuration file. However, eTrust CA Top Secret mandates that sites only use up to the first eight characters in the IBMFAC resource class. Therefore, you could define the IBMFAC entry as follows:

Example 1

```
TSS ADD(SYSADM) IBMFAC($SKTVIEW)
```

Many sites have a requirement that personnel have authority levels appropriate for both production and development CICS regions. The PERMIT command allows one to place the full 44-character entity name for an IBMFAC entry.

The default access of NONE prevents users from using the Unicenter NetMaster command/control server.

READ access enables users to display session, server and global statistics.

UPDATE access allows users to start and stop transactions, sessions, servers and applications.

Once the entities are defined in the FACILITY resource class, you must give personnel the proper authority to do their job.

The following are sample FACILITY entity definitions for use by MVS system personnel (SYS1), where:

CICSOPR CICS operator.

CICSDEV A CICS developer.

\$SKTVIEW.CICSPROD.COMDAUTH

Configured in the SECENT parameter for the CICS production regions.

\$SKTVIEW.CICSTEST.COMDAUTH

Configured in the SECENT parameter for the CICS development regions.

Example 2

```
TSS PERMIT(SYS1,CICSOPR)  IBMFAC($SKTVIEW.CICSPROD.COMDAUTH)  ACCESS(UPDATE)
TSS PERMIT(SYS1,CICSOPR)  IBMFAC($SKTVIEW.CICSTEST.COMDAUTH)  ACCESS(UPDATE)
TSS PERMIT(CICSDEV)       IBMFAC($SKTVIEW.CICSPROD.COMDAUTH)  ACCESS(READ)
TSS PERMIT(CICSDEV)       IBMFAC($SKTVIEW.CICSTEST.COMDAUTH)  ACCESS(UPDATE)
```

- Required TSS APARs
- For command authority security checking to work properly, you must:
- Install eTrust CA-Top Secret APAR KEC5385.
 - You must also specify "OPTIONS(70)" in the TSS startup parameter file and recycle TSS. KEC5385 allows the CICS principal facility (terminal) associated with the command server to be defined as an output-only device (see topic "Define the principal facility to be associated with T09TCMDS").

Note: The "OPTIONS(70)" parameter causes KEC5385 to be dynamically implemented. Without this APAR, command server authorization is essentially bypassed, and full command authority will be granted to all users with valid user IDs and passwords that sign on to the command server.

Contact TSS technical support for details and assistance in obtaining and installing KEC5385.

Security Configuration for RACF

Many sites have both production and test CICS regions. You should give operators and system personnel the proper authority to do their job. At the same time, you want to protect the CICS applications from overly inquisitive or unruly personnel in the user community.

The 44 characters resource name must match the SECENT parameter of the T09MCMDS macro in the T09CONxx configuration file. Many sites have a requirement that personnel have authority levels appropriate for both production and development CICS regions.

Here are two sample FACILY entity definitions for use by a production and a CICS region:

Example 1

```
RDEFINE FACILITY $SKTVIEW.CICSPROD.COMDAUTH UACC(NONE)
RDEFINE FACILITY $SKTVIEW.CICSTEST.COMDAUTH UACC(NONE)
```

The default access, NONE, prevents users from utilizing the Unicenter NetMaster command/control server.

READ access enables users to display session, server and global statistics.

UPDATE access enables users to start and stop transactions, sessions, servers and applications.

Once the entities are defined in the FACILITY resource class, you want to give personnel the proper authority to do their job.

The following are sample FACILITY entity definitions for use by MVS system personnel (SYS1), where:

CICSOPR A CICS operator.

CICSDEV A CICS developer.

\$SKTVIEW.CICSPROD.COMDAUTH

 Configured in the SECENT parameter for the CICS production regions.

\$SKTVIEW.CICSTEST.COMDAUTH

 Configured in the SECENT parameter for the CICS development regions.

Example 2

```
PERMIT $SKTVIEW.CICSPROD.COMDAUTH CLASS(FACILITY) ACCESS(UPDATE) ID(SYS1,CICSOPR)
PERMIT $SKTVIEW.CICSPROD.COMDAUTH CLASS(FACILITY) ACCESS(READ) ID(CICSDEV)
PERMIT $SKTVIEW.CICSTEST.COMDAUTH CLASS(FACILITY) ACCESS(UPDATE) ID(SYS1,CICSOPR)
PERMIT $SKTVIEW.CICSTEST.COMDAUTH CLASS(FACILITY) ACCESS(UPDATE) ID(CICSDEV)
```


Diagnostic Commands

This chapter describes the diagnostic commands available.

It includes the following topics:

- [TCPEEP](#) – A command to invoke the packet trace program and diagnose remote host communication problems
- [TRACE](#) – Trace command enhancements to collect TCP/IP data and display it on a terminal or send it to an external writer

TCPEEP

TCPEEP is a TSO command that invokes the packet trace program to diagnose remote host communication problems. The TCPEEP real time trace consists of selected network packet traffic to and from a local host. The TCPEEP command recognizes many levels of network traffic, however we are only addressing CICS/TS socket tracing in this chapter.

TCPEEP creates a NO WRAP Component Trace Instance and displays the output on a TSO terminal or directs it to a dynamically allocated SYSOUT data set. Optionally, it can stop any component trace instance, or modify an existing component trace instance or view an existing Component Trace Instance.

Note: TCPEEP runs only when the TRACE address space is active.

User Interface

TCPEEP can be run as a TSO command, either from TSO or as a batch TSO.

The JCL to run TCPEEP as a batch job can be copied from the T09SAMP library member T09PEEP. TRACE must be up and running before submitting a batch job for TCPEEP.

The following is a sample JCL for running TCPEEP in batch.

```
//T09PEEP JOB (TCPEEP), 'TCPEEP', CLASS=A, MSGCLASS=X
/*
/* SAMPLE JCL TO RUN TCPEEP IN BATCH.
/*
/* UPDATE 'TRGINDX' TO REFLECT YOUR LIBRARY NAMING CONVENTION.
/*
/* NOTE: THE TRACE ADDRESS SPACES MUST BE RUNNING.
/*
/* NOTE: IF UNICENTER TCPACCESS, THEN IT TOO MUST BE RUNNING.
/*
/*TCPEEP EXEC PGM=IKJEFT01, DYNAMNBR=50, REGION=4M
/*STEPLIB DD DISP=SHR, DSN=TRGINDX.LINK
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*, LRECL=132, BLKSIZE=132, RECFM=FB
//SYSIN DD DUMMY
/*
/* USE THE FOLLOWING TCPEEP FOR UNICENTER TCPACCESS INSTALLATIONS
/*
//SYSTSIN DD *
TCPEEP TRCSSID(ACTR) FORMAT(TCP,DATA(16),IUCVDATA(16),EZADATA(16)) +
GROUPS((EZA,'MAXEZADATA(16)'), +
(IUCV,'MAXIUCVDATA(16)'))
//
//
/*
/* USE THE FOLLOWING TCPEEP FOR IBM TCP/IP INSTALLATIONS
/*
/*SYSTSIN DD *
TCPEEP TRCSSID(ACTR) FORMAT(EZADATA(16)) +
GROUPS((EZA,'MAXEZADATA(16)'))
/*
```

To stop the TCPEEP batch job, issue the MVS **STOP** command (for example, **P jobname**).

Trace Data Collected

The trace facility collects the following type of trace data:

- From an application
- Unicenter NetMaster Socket Management for CICS
- Optionally from the TCP/IP address space, if you are using Unicenter TCPaccess

Simultaneous tracing of various trace types can be fed into a single trace output.

Note: If data is collected via the external writer to an external data set, you must use the MVS TRACE command. See [External Writer](#) for more information.

Viewing Trace Data

TCPEEP collects realtime data and by default writes to SYSTSPRT. SYSTSPRT, your TSO screen. Optionally, it can write to a dynamically allocated SYSOUT data set, so that it can be viewed in the JES spool.

Trace Operation

Each occurrence of the TCPEEP command varies according to the type of environment.

To stop a trace, press the terminal ATTENTION key and enter **H** at the prompt (null entry allows trace to continue).

WARNING! *TCPEEP should be installed in a protected library only. It can be used to display all network traffic through Socket Management, and Unicenter TCPaccess, including user IDs, passwords, or even payroll information if it goes out on a network.*

TCPEEP Syntax

This section describes the TCPEEP syntax and describes its parameters.

```
TCPEEP [ ASID( asid,... ) ][ BUFFERS ( size, number ) ][ BUFFTME ( time_out ) ]  
[ DATASIZE( record_size ) ][ FORMAT( format_options ) ]  
[ FULL | SUMMARY ][ GROUPS( ( group [, 'filter' ) ... ) ][ HALT ]  
[ INSTANCE ( instance_ID ) ][ JOBNAME ( jobname,... ) ][ NOHEADER ]  
[ PEEK ( limit ) ][ SYSOUT ( class ) ][ TRACE SIZE ( num_records ) ]  
[ TRCSSID ( ssid ) ]
```

Note: Also refer to the “Diagnostic Commands” chapter of *Unicenter TCPaccess Communications Server System Management Guide* for other TCPEEP options that can be coded together with the Socket Management options.

ASID (*asid*, ...)

Specifies the address space identifiers (ASIDs) of address spaces used as a filter for tracing. Events in the ASIDs are recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas.

An empty ASID list, ASID=(), turns off filtering by address spaces. In the ASID parameter, list all address spaces to trace. Address spaces for previous traces are not traced unless listed.

Default: None.

BUFFERS (*size*, *number*)

Specifies the size of the trace buffers in kilobytes or the number of buffers.

size A value between 64 and 1024 (Default: 256).

num A value between 2 and 128 (Default: four).

BUFFERS (Optional). Can only be specified when creating a new trace instance. If specified when modifying a trace instance, it is ignored.

Range: *Size*: 64 - 1024
 Number: 2 - 128.

Default: (256, 4).

BUFFTIME (*time_out*) Specifies the buffer time out interval in seconds. At the end of each interval, if the current buffer contains data but is not full, a buffer flush operation is initiated.

BUFFTIME is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

Use this parameter to force a buffer switch so you do not have to wait for the entire buffer to fill to see trace data.

Range: 0 – 99999.

Default: 10.

DATASIZE (*record_size*)

Specifies the maximum size of a trace record in kilobytes. Trace records that exceed the specified value are truncated.

DATASIZE – (Optional). Can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

If the specified maximum size exceeds the largest supported trace record size (64 KB less control headers), then the specification has no effect.

Default: None.

FORMAT (*format_options*)

The format of the collected data.

Formatted data can be specified as follows:

EZADATA | EDATA | EZAEBDIC | EZAASCII

or

EZADATA(*nn*) | EDATA(*nn*) | EZAEBDIC(*nn*) | EZAASCII(*nn*)

- Where *nn* is between 0 and 65535 (default is 65535)

Note: A data amount set very high generates a very large trace in a short period. A high value should be avoided unless you are absolutely convinced that there is a data integrity problem. For the vast majority of problems a glance value of 16 is sufficient. A value of 16 causes one line of data to display.

- EZADATA | EDATA uses a mixed ASCII and EBCDIC translate table
- EZAEBDIC | EEBCDIC uses an EBCDIC-only translate table
- EZAASCII | EASCII uses an ASCII-only translate table

FULL | SUMMARY Show all or a summary of the data.

Default: SUMMARY.

GROUPS((*group* [, '*filter*']) ...)

Trace group or groups for which data is collected and optionally a filter parameter for each group.

Use the GROUPS parameter to limit the amount of data collected. Used with INSTANCE to modify an existing trace instance, either adds a new group to the trace instance or replaces an existing group for the trace instance. Once added, a group cannot be removed from the active trace instance.

For Sockets Management only group EZA is supported. Refer to the “Diagnostic Commands” chapter in the *Unicenter TCPaccess Communications Server System Management Guide* for other TCPEEP group options that can be coded together with the Socket Management options.

You can specify a maximum of four trace groups.

group Selects the type of data to collect: EZA

For TCPaccess only: IUCV, NETIF.

The EZA is a collection of trace points in Socket Management for processing EZASOCKET API activity with the following filter options:

HOST(*host,...,host*) – Up to 16 IP HOST addresses (names).

PORT(*port,...,port*) – Up to 16 port numbers (names).

USER(*jobname,..,jobname*) – Takes 1 to 16 *jobnames* that refer to jobs using the EZA function.

UASID(*asid,...,asid*) – Takes 1 to 16 *asids* that refer to jobs using the EZA function.

MAXEZADATA(*nnnn*) | MEZADATA(*nnnn*) | MEDATA(*nnnn*) – Where *nnnn* is a positive integer less than or equal to 65535.

Notes:

- A data amount set very high generates a very large trace in a short period. A high value should be avoided unless you are absolutely convinced that there is a data integrity problem. For the vast majority of problems a glance value of 16 is sufficient. A value of 16 causes one line of data to display.
- By default, no user data is traced, so MAXEZADATA causes the data to be collected.

It is better to limit EZA GROUP activity through the USER() and UASID() parameters rather than the PORT() and HOST() parameters. The PORT() and HOST() parameters require that session to be established thus one will loose all the API commands and output up to the point when a session has been established.

HALT Stops a component trace instance.

INSTANCE (*instance_id*)

Select a trace instance to display by specifying the *instance_id* returned from TCPEEP or the MVS TRACE CT command.

JOBNAME (*jobname ,...*)

Names of jobs used as filters for tracing. Events in these jobs are recorded by the component trace.

NOHEADER

Do not display the header information from the trace entry.

This option helps limit the output. Without it, one line prints describing the trace entry even if no other information about that entry displays.

PEEK (*limit*)

Number of trace records to view.

Use to view an existing trace. PEEK or PEEK(zero) implies no limit; PEEK(*n*) traces only *n* trace records.

Range: Zero - no limit.

Default: Zero.

SYSOUT (*class*)

Sends output to a dynamically allocated SYSOUT data set.

By default, the output for TCPEEP writes to SYSTSPRT.

Default: X.

TRACESIZE (*num_records*)

Maximum number of trace records to be recorded. If not specified, there is no limit to the number of records recorded.

TRACESIZE is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

TRCSSID (*ssid*)

ID of the trace address space.

Default: ACTR.

Useful TCPEEP Commands

This section lists the keywords for each function listed.

Creating a NOWRAP Trace

Use the following command to initiate a NO WRAP trace instance:

```
TCPEEP TRCSSID( trcsubsysid ) GROUPS( group )           +
      JOBNAME( jobname ) ASID( asid ) DATASIZE( size ) TRACESIZE( trcsize ) +
      BUFFERS( bufname ) BUFFTME( buftime ) FORMAT( format_options )      +
      SYSOUT( class )
```

Stopping Trace

Use this command to stop any trace instance:

```
TCPEEP INSTANCE( inst ) HALT TRCSSID( trcsubsysid )
```

Modify Existing Trace

Use this command to modify an existing trace instance:

```
TCPEEP INSTANCE( inst ) TRCSSID( trcsubsysid ) DNRSSID( dnrsubsysid )      +
      GROUPS ( group ) JOBNAME( jobname ) ASID( asid )      PEEK          +
      FORMAT( format_options ) SYSOUT( class )
```

Regardless of how the trace is started, view its LIFO for the specified number of records (PEEK or PEEK(0) implies no limit).

View Existing Trace

Use this command to view an existing trace instance:

```
TCPEEP INSTANCE( inst ) TRCSSID( trcsubsysid PEEK      +
      FORMAT( format_options ) SYSOUT
```

Regardless of how the trace is started, view its LIFO for the specified number of records (PEEK or PEEK(0) implies no limit).

Specifying GROUPS

This example shows how to specify the GROUPS for TCPEEP.

The following example allows you to specify EZA without additional filtering.

```
TCPEEP GROUPS ( EZA )
```

To specify EZA and MEDATA(32) :

```
TCPEEP GROUPS ( ( EZA, ' MAXEZADATA ( 32 ) ' ) )
```

The following specifies an EZA and an IUCV group. You may use the IUCV keyword to specify an IUCV or HPNS application for tracing.

Note: This example only works for and is the recommend parameters for TCPaccess installations.

```
TCPEEP TRCSSID(ACTR) FORMAT(TCP,DATA(32),IUCVDATA(32),EZADATA(32)) +
      GROUPS((EZA, 'MAXEZADATA(32)'), +
      (IUCV, 'MAXIUCVDATA(32)'))
```

TCPEEP Example

This following section shows an excerpt of the output from a TCPEEP. This trace shows a small portion of a connection sequence from a loopback client into a CICS server. This is **not** a complete trace and has been edited so that it could be included here. You will get similar results but not the exact entries when you run a trace. This is only meant as an example and is not all-inclusive. This excerpt from the trace includes the following sequence of trace events:

- Server finishing issuing a listen call
- Server issuing a select call waiting for connections
- Client issuing an initapi call
- Client issuing a socket call
- Skip of Client issuing get type calls
- Client issuing a connect call

Note: The trace parameters are displayed at the beginning of the trace.

```

READY
  TCPEEP TRCSSID(ACQA) FORMAT(TCP,DATA(16),IUCVDATA(16),EZADATA(16))
GROUPS((EZA,'MAXEZADATA(16)') (IUCV,'MAXIUCVDATA(16)')
)
T03PE010I TRACE INITIATED - INSTANCE: 01
-----
SOK-EXIT 000B1857 04/02 12:52:55.737764 EZA - EZASOKET Function Complete
0823 LONCICS5 Cmpl LISTEN TIME OUT 04/02/2002 12:52:55.737753
  Conn: Prot(TCP) Local=0.0.0.0:1234 Remote=0.0.0.0:0
  EZASOKET 151E6B65 1540091E 1540092C 15400924
  EZASOKET 95400928
  RETCODE(0) S(0) BACKLOG=5
  TRAN=IPTL TASK=36 S(0) USERID=MEL2
  TOKEN=00000005 CEP=153E8008 HIW=153DFC88 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F6C3
-----
SOKENTER 000B0857 04/02 12:52:55.737818 EZA - EZASOKET Function Enter
0823 LONCICS5 Entr SELECT 12:52:55.737818
  Conn: Prot(TCP)
  EZASOKET 151E6B85 15400930 151E6BD4 154009B8
  EZASOKET 15400AB8 15400BB8 15400CB8 15400DB8
  EZASOKET 15400EB8 15400924 95400928
  MAXSOC=50 TIMEOUT: SEC(-1) USEC(-1)
  RSNDSK 00000001 00000000 000A
-----
SOKENTER 000B0857 04/02 12:54:13.593463 EZA - EZASOKET Function Enter
0823 LONCICS5 Entr INITAPI TIME IN 04/02/2002 12:54:13.593357
  EZASOKET 00197B97 15418034 1541805C 15418038
  EZASOKET 15418070 15418024 95418020
  MAXSOCK=256 IDENT= Subtask=F0F0F0F0F0F3F8E2
  TRAN=JSC1 TASK=38 S(-1) USERID=MEL2 TERMINAL=A03VLT36
  TOKEN=00000006 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2

```

```
-----
SOK-EXIT 000B1857 04/02 12:54:13.826950 EZA - EZASOKET Function Complete
0823 LONCIC55 Cmpl INITAPI TIME OUT 04/02/2002 12:54:13.826919
EZASOKET 00197B97 15418034 1541805C 15418038
EZASOKET 15418070 15418024 95418020
RETcode(0) MAXSOCK=256 IDENT= Subtask=F0F0F0F0F0F3F8E2
MAXSNO=255
TRAN=JSC1 TASK=38 S(-1) USERID=MEL2 TERMINAL=A03VLT36
TOKEN=00000006 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2
-----
SOKENTER 000B0857 04/02 12:54:13.827098 EZA - EZASOKET Function Enter
0823 LONCIC55 Entr SOCKET TIME IN 04/02/2002 12:54:13.827063
EZASOKET 00197A90 15417E90 15417E94 15417E98
EZASOKET 15417E64 95417E60
SOCTYPE(TCP) NS=-1
TRAN=JSC1 TASK=38 S(-1) USERID=MEL2 TERMINAL=A03VLT36
TOKEN=00000006 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2
-----
SOK-EXIT 000B1857 04/02 12:54:13.829499 EZA - EZASOKET Function Complete
0823 LONCIC55 Cmpl SOCKET TIME OUT 04/02/2002 12:54:13.829481
Conn: Prot(TCP) Local=0.0.0.0:0 Remote=0.0.0.0:0
EZASOKET 00197A90 15417E90 15417E94 15417E98
EZASOKET 15417E64 95417E60
RETcode(0) SOCTYPE(TCP) NS=-1
TRAN=JSC1 TASK=38 S(0) USERID=MEL2 TERMINAL=A03VLT36
TOKEN=00000007 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2
-----
Skipped>>>>0823 LONCIC55 Entr GETHOSTBYNAME 12:54:13.829660
Skipped>>>>0823 LONCIC55 Entr GETHOSTID TIME IN 04/02/2002 12:54:14.044875
Skipped>>>>0823 LONCIC55 Entr GETHOSTBYADDR TIME IN 04/02/2002 12:54:14.045520
-----
SOKENTER 000B0857 04/02 12:54:14.069114 EZA - EZASOKET Function Enter
0823 LONCIC55 Entr BIND TIME IN 04/02/2002 12:54:14.069047
Conn: Prot(TCP) Local=141.202.36.71:0 Remote=0.0.0.0:0
EZASOKET 00197726 154121B2 15417E98 15417E6C
EZASOKET 95417E68
S(0) NAME: Family(2) Port=0 IP=0.0.0.0
TRAN=JSC1 TASK=38 S(0) USERID=MEL2 TERMINAL=A03VLT36
TOKEN=00000007 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2
-----
SOK-EXIT 000B1857 04/02 12:54:14.069272 EZA - EZASOKET Function Complete
0823 LONCIC55 Cmpl BIND TIME OUT 04/02/2002 12:54:14.069262
Conn: Prot(TCP) Local=0.0.0.0:3055 Remote=0.0.0.0:0
EZASOKET 00197726 154121B2 15417E98 15417E6C
EZASOKET 95417E68
RETcode(0) S(0) NAME: Family(2) Port=0 IP=0.0.0.0
TRAN=JSC1 TASK=38 S(0) USERID=MEL2 TERMINAL=A03VLT36
TOKEN=00000007 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2
-----
SOKENTER 000B0857 04/02 12:54:14.069364 EZA - EZASOKET Function Enter
0823 LONCIC55 Entr CONNECT TIME IN 04/02/2002 12:54:14.069345
Conn: Prot(TCP) Local=0.0.0.0:3055 Remote=0.0.0.0:0
EZASOKET 00197756 154121B2 15417E98 15417E6C
EZASOKET 95417E68
S(0) NAME: Family(2) Port=1234 IP=141.202.36.71
TRAN=JSC1 TASK=38 S(0) USERID=MEL2 TERMINAL=A03VLT36
TOKEN=00000007 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71
Subtask=F0F0F0F0F0F3F8E2
```

```
-----  
SOK-EXIT 000B1857 04/02 12:54:14.069614 EZA - EZASOKET Function Complete  
0823 LONCIC55 Cmpl CONNECT TIME OUT 04/02/2002 12:54:14.069602  
Conn: Prot(TCP) Local=0.0.0.0:3055 Remote=141.202.36.71:1234  
EZASOKET 00197756 154121B2 15417E98 15417E6C  
EZASOKET 95417E68  
RETCODE(0) S(0) NAME: Family(2) Port=1234 IP=141.202.36.71  
TRAN=JSC1 TASK=38 S(0) USERID=MEL2 TERMINAL=A03VLT36  
TOKEN=00000007 CEP=153EB008 HIW=153E8D68 TCPIP=TCPIP71  
Subtask=F0F0F0F0F0F3F8E2
```

TRACE

The IBM MVS Component Trace facility has been enhanced to provide a method of collecting TCP/IP data and displaying it on a terminal or sending it to an external writer. Additional JCL is required to enable the Component Trace enhancements. To use Component Trace as a TSO command, see [TCPEEP](#).

MVS Component Trace

MVS Component Trace is a diagnostic aid used to trace the action of certain system components and third party components that define themselves to Component Trace. The TRACE operator command is used to start, stop and control the component trace. For more information on the MVS TRACE command, see the IBM MVS System Commands.

Trace Address Space

Component Trace is defined in its own address space and collects trace data for trace points defined in other address spaces. Likewise, the other address spaces must identify the Component Trace address space for the data collection. Multiple occurrences of Component Trace can be active at the same time, each with a unique subsystem ID.

If you are running Unicenter TCPaccess, for each address space, IJTFCGxx in the PARM member defines the subsystem ID. The definition is specified using the TRACENAME keyword on the IFSPARM statement as follows:

```
IFSPARM PROMPT VMCFNAME( VMCF ) TRACENAME( ACTR ) NOPROMPT
```

In the example, the trace data in the address space is directed to the Component Trace address space with an ACTR subsystem ID.

Note: If the Trace Address Space is brought up after the TCP stack, there may be a delay of two minutes before events are traced.

Exit

MVS Component Trace requires an exit to communicate with the tracing component. The T03PTRSS exit must reside in LPALIB or the Link List. It is distributed in the LINK library.

External Writer

The collected trace data is written to DASD or TAPE using an external writer. A suitable External Writer Cataloged Procedure for use with Component Trace may already be defined on your system. For more information on defining an External Writer used with Component Trace, see the IBM publication *MVS Diagnosis: Tools and Service Aids*.

Component Trace JCL

The JCL to run a Component Trace Address Space can be copied from the T09SAMP library member T09TRACE. The following is a sample JCL for running trace.

CAUTION! Be sure to specify *P=T03* (shown in bold below) on the TRACE statement to specify the Component Trace Address space. If you specify T01, you will bring up a second Unicenter TCPaccess stack.

```
//T09TRACE JOB
//*
//* SAMPLE JCL PROCEDURE TO RUN SOCKET MANAGEMENT TRACE ADDRESS SPACE
//* NOTE: THIS ADDRESS SPACE SHOULD NOT BE TERMINATED
//*
//* EDIT THE TRGINDX, SSN, SOUT, CMND SYMBOLIC PARAMETERS
//*
//* VERIFY THAT THE JOB CARD AND NAMING CONVENTIONS MEET
//* YOUR SITE'S JCL REQUIREMENTS, THEN SUBMIT THIS JOB.
//*
//ICSTRACE PROC TRGINDX='TRGINDX', TARGET LIBRARIES DSN INDEX
//          SSN=ACTR,          DFLT SUBSYSTEM NAME
//          SOUT='*',          CHOOSE A HOLD NONPURGE SYSOUT CLASS
//          CMND=STARTTR,      DFLT STARTUP COMMAND SCRIPT NAME
//          CNFG=00           IJTFCGXX SUFFIX
//*
//TRACE EXEC PGM=IFSSTART,REGION=6144K,TIME=1440,
// PARM=' IFSINIT,U=&SSN,P=T03,SO=&SOUT,CM=&CMND,CF=&CNFG'
//*
```

```
//STEPLIB DD DISP=SHR,DSN=&TRGINDX..LOAD
//*
/* WARNING: THE LOAD DATA SET MUST NEVER BE ADDED TO THE LINK LIST.
/* COMPONENT TRACE ELEMENT NAMES ARE NOT UNIQUE AND MAY
/* AFFECT THE OPERATIONS OF OTHER SOFTWARE.
/* THE LIBRARY SHOULD ALWAYS BE REFERENCED THROUGH
/* A STEPLIB OR JOBLIB STATEMENT.
/*
/* CONFIGURATION DATA SETS
/*
//SYSPARM DD DISP=SHR,DSN=&TRGINDX..PARM
//SYSPROC DD DISP=SHR,DSN=&TRGINDX..PARM
//SYSHELP DD DISP=SHR,DSN=&TRGINDX..HELP
/*
/* LOG DATA SETS
/*
//T01LOG DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT
/*
/* DUMP DATA SETS
/*
//SYSUDUMP DD SYSOUT=&SOUT
/*
/* MISC DATA SETS
/*
//ABNLIGNR DD DUMMY /* DISABLE ABEND-AID PROCESSING */
/*
```

TRACE Command

The TRACE command starts, stops, modifies, or displays the status of a system trace, master trace, or component trace.

The TRACE command is a standard MVS operator command.

```
TRACE [ CT { [ , ON | OFF ] [ , COMP=name ] [ , PARM=mem ] } ]
      [ CT { [ , WTRSTART=mem_name[,WRAP | NOWRAP ] ] } ]
      [ CT { , WTRSTOP=job_name } ] ]
```

- | | |
|-------------------|---|
| ON | Turns on tracing for a component if the component trace is currently off. If the component trace is on and can be changed, this changes the trace options. |
| OFF | Turns off tracing for the component. If the component is connected to an external writer, the trace is implicitly disconnected from the writer. |
| COMP= <i>name</i> | Identifies the component trace with the subsystem ID for the trace address space. This is required for each TRACE command. |
| PARM= <i>mem</i> | Identifies a member of SYS1.PARM or a data set in the system parmlib concatenation containing the parameters used for tracing. Using a parmlib member enables the operator to initiate the trace, change it, or stop it without a message prompting for parameters. |

Parameters specified on the TRACE command override the options specified in the parmlib member. The parameters are described in [TRACE Command Reply](#).

WTRSTART= *mem_name*

Identifies the member containing the JCL to invoke an external writer and opens the data sets used by the external writer. The member must be a SYS1.PROCLIB cataloged procedure or a job.

After starting the external writer, use the WTR parameter to connect the component trace to the external writer.

WRAP | NOWRAP

NOWRAP instructs the system to stop writing data to a data set when the data set is full. With the WRAP parameter, when the data set or group of data sets is full, new data overwrites the oldest data at the start of the data set or the start of the first data set.

If the WTRSTART parameter on the TRACE CT command specifies NOWRAP, the system uses the primary and secondary extents of the data set or sets. If the WTRSTART parameter specifies WRAP or omits the parameter, the system uses only the primary extent or extents.

WTRSTOP=*job_name*

Disconnects the external writer from the component trace and closes the data sets used by the external writer.

jobname is the member name if the source JCL is a procedure, or a job name if defined on a JOB statement within the source JCL.

Before stopping the external writer, turn the component trace off with **TRACE CT,OFF** or disconnect the external writer with **WTR=DISCONNECT**.

TRACE Command Reply

In response to a TRACE CT,ON command without the PARM parameter, the system prompts you to specify the component trace options. Use the REPLY command to respond.

```
R id[ ,ASID=( nnnn [ ,nnnn ]... ) ]  
  [ ,CONT | ,END ]  
  [ ,JOBNAME=( name [ ,name ]... ) ]  
  [ ,OPTIONS=( option [ ,option ]... ) ]  
  [ ,WTR={ mem_name | DISCONNECT } ]
```

id Use the same identification number (0-9999) from the message to identify the reply.

ASID=(nnnn [, nnnn] ...)

Specifies the address space identifiers, ASIDs of address spaces used as a filter for tracing. Events in the ASIDs are recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas. An empty ASID list, ASID=(), turns off filtering by address spaces.

In the ASID parameter, list all address spaces to be traced. Address spaces for previous traces are not traced unless listed.

CONT or END

The CONT parameter continues the reply on another line. The system issues another reply message. You can then continue the reply and repeat any parameters on the continuation line, except END. Repeated parameters are strung together, they do not overlay each other. The END parameter identifies the end of the REPLY.

CONT or END must be the last parameter on the input line.

JOBNAME=(name [, name] ...)

Names of jobs used as filters for tracing. Events in these jobs are recorded by the component trace.

The parameter contains a list of 0 to 16 job names separated by commas.

An empty job list, JOBNAME=(), turns off filtering by jobs.

In the JOBNAME parameter, list all jobs to be traced. Jobs specified for previous traces are not traced unless listed.

OPTIONS=(option [, option] ...)

Specifies the component trace options described in [TRACE Command Reply Options](#).

WTR= *mem_name* | DISCONNECT

The *membername* identifies the member containing the source JCL that invokes the external writer. The member must be a SYS1.PROCLIB cataloged procedure or a job. The membername in the WTR parameter must match the membername in the TRACE CT,WTRSTART command.

WTR=DISCONNECT disconnects the external writer. The component continues tracing and placing the trace records in the address-space buffer, but stops passing trace records to the external writer.

You must also specify a TRACE CT,WTRSTART or TRACE CT,WTRSTOP command to start or stop the writer.

TRACE Command Reply Options

Use the Options parameter in response to the Reply prompt.

```
OPTIONS= [ BUFFERS( size,num ) ] [ BUFFTME( time ) ] [ DATASIZE( dsize ) ]
          [ GROUPS( ( group[, 'filter' ] )...) ] [ HALT ]
          [ INSTANCE( inst ) ] [ STATUS ] [ TRACE SIZE( tsize ) ] [ WRAP ]
          [ NOWRAP ]
```

BUFFERS (*size,num*) Size of the trace buffers in kilobytes or the number of buffers.

size A value between 64 and 1024.

Default: 256.

num A value between 2 and 128.

Default: Four.

BUFFERS can only be specified for a new trace instance. BUFFERS is optional and should only be specified when creating a trace instance.

Note: If specified when modifying a trace instance, it is ignored.

BUFFTME(*time*) Buffer timeout interval in seconds. At the end of each interval, if the current buffer contains data but is not full, a buffer flush operation is initiated.

BUFFTME is optional and is considered only when creating a trace instance.

Use this parameter to force a buffer switch so you do not have to wait for the entire buffer to fill to see trace data.

Range: 0 - 99999.

Default: 10.

DATASIZE(*dsize*) Specifies the maximum size of a trace record in kilobytes. Trace records that exceed the specified value are truncated.

DATASIZE is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored. If the specified maximum size exceeds the largest supported trace record size (64 KB less control headers), then the specification has no effect.

Default: Any size record is recorded (up to the maximum IBM limit of 64 KB less control headers.)

GROUPS((*group* [, 'filter']) ...)

Trace group or groups for which data is collected and optionally a filter parameter for each group.

Use the GROUPS parameter to limit the amount of data collected. Used with INSTANCE to modify an existing trace instance, either adds a new group to the trace instance or replaces an existing group for the trace instance. Once added, a group cannot be removed from the active trace instance.

For Sockets Management only group EZA is supported. Refer to the “Diagnostic Commands” chapter in the *Unicenter TCPaccess Communications Server System Management Guide* for other TCPEEP group options that can be coded together with the Socket Management options.

You can specify a maximum of four trace groups.

group Selects the type of data to collect: EZA

For TCPaccess only: IUCV, NETIF.

The EZA is a collection of trace points in Socket Management for processing EZASOCKET API activity with the following filter options:

HOST(host,...,host) – Up to 16 IP HOST addresses (names).

PORT(port,...,port) – Up to 16 port numbers (names).

USER(jobname,...,jobname) – Takes 1 to 16 jobnames that refer to jobs using the EZA function.

UASID(asid,...,asid) – Takes 1 to 16 asids that refer to jobs using the EZA function.

MAXEZADATA(nnnn) | MEZADATA(nnnn) | MEDATA(nnnn) – Where nnnn is a positive integer less than or equal to 65535.

Notes:

- A data amount set very high generates a very large trace in a short period. A high value should be avoided unless you are absolutely convinced that there is a data integrity problem. For the vast majority of problems a glance value of 16 is sufficient. A value of 16 causes one line of data to display.
- By default, no user data is traced, so MAXEZADATA cause s the data to be collected.

It is better to limit EZA GROUP activity through the USER() and UASID() parameters rather than the PORT() and HOST() parameters. The PORT() and HOST() parameters require that session to be established or you will loose all the API commands and output up to the point when a session was established.

HALT	<p>Specifies that the trace instance identified by the INSTANCE keyword should be stopped.</p> <p>If HALT is specified, INSTANCE must be specified. If the INSTANCE is connected to an external writer, WTR=DISCONNECT must also be specified.</p>
INSTANCE (<i>inst</i>)	<p>Modifies or stops a trace instance.</p> <p><i>inst</i> is the component trace address space ID. A new trace instance is created if <i>inst</i> is not specified.</p>
STATUS	<p>Displays all active trace instances.</p> <p>Note: If STATUS is specified, any other keywords are ignored.</p>
TRACESIZE(<i>tsize</i>)	<p>Maximum number of trace records to record. If not specified, there is no limit to the number of records recorded.</p> <p>TRACESIZE is optional and can only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.</p>

WRAP	<p>Create an in-memory WRAP trace.</p> <p>Not valid with <code>WTR=mem_name</code>, nor can the instance be modified later to specify <code>WTR=mem_name</code>. This is the default if <code>NOWRAP</code> is not specified and <code>WTR=mem_name</code> is omitted.</p>
NOWRAP	<p>Creates a NOWRAP trace. This is the default if <code>WTR=mem_name</code> is specified. The external writer writes the buffers when they fill up and after being written are reused.</p> <p>This option is for the Trace Address space and should not be confused with <code>WRAP NOWRAP</code> on the <code>TRACE CT</code> command, which is for the IBM writer.</p>

Comparing TCPEEP and MVS TRACE Syntax

The syntax for commands using TCPEEP differ from those used with the MVS TRACE facility.

OPTIONS To use MVS TRACE to specify EZA and MEDATA(32), you need to specify:

```
OPTIONS = ( GROUPS ( ( EZA, ' MAXEZADATA ( 32 ) ' ) ) )
```

This is the same as the TSO TCPEEP command:

```
TCPEEP GROUPS ( ( EZA, ' MAXEZADATA ( 32 ) ' ) )
```

JOBNAME To specify JOBNAME using the MVS TRACE command, the syntax is:

```
JOBNAME = ( jobname1, ..., jobname16 )
```

When using TCPEEP the syntax is

```
TCPEEP JOBNAME ( jobname1, ..., jobname16 )
```

For more information about TCPEEP, see [TCPEEP](#).

Trace Command Examples

The following examples demonstrate the control of the T03 Trace Facility using the MVS TRACE command. The examples assume a T03 Trace Facility is active and is using a Subsystem ID of ACTR.

Starting a Trace Instance

In this example a new trace instance is created. The instance is limited to tracing 10000 records for group ID EZA from address space CICSPROD.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( GROUPS(EZA ) TRACESIZE( 10000 ) ),JOBNAME=(CICSPROD),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR909I Trace start successful Instance(01)
```

Modifying a Trace Instance

In this example an existing trace instance is modified. The trace instance is changed to include records for group ID EZA.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( INSTANCE( 1 ) GROUPS( EZA ) ),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Displaying Trace Status

In this example the status of the trace instances is displayed.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( STATUS ),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR916I Instance(01) Active, records=23,745
T03TR916I Instance(02) Active, records=1,576
```

Stopping a Trace Instance

In this example an existing trace instance is stopped.

```
TRACE CT,ON,COMP=ACTR
R xx,OPTIONS=( INSTANCE( 1 ) HALT ),END
```

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR910I Trace shutdown successful Instance(01)
```

Stopping All Trace Instances

In this example all existing trace instances are stopped.

```
TRACE CT,OFF,COMP=ACTR
T03TR910I Trace shutdown successful Instance(01)
T03TR910I Trace shutdown successful Instance(02)
```

Starting an External Writer

In this example an external writer is started. T03XWTR is the name of a predefined started task.

```
TRACE CT,WTRSTART=T03XWTR
```

Starting a Trace Instance and Connecting an External Writer

In this example a new trace instance is created and connected to an external writer. Only one instance can have an external writer connected.

```
TRACE CT,ON,COMP=ACTR  
R xx,OPTIONS=( GROUPS( NETIF ) ),WTR=T03XWTR,END
```

Where *xx* IIT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR909I Trace start successful Instance(01)
```

Modifying a Trace Instance to Connect an External Writer

In this example an existing trace instance is connected to an external writer. Only one instance can have an external writer connected.

```
TRACE CT,ON,COMP=ACTR  
R xx,OPTIONS=( INSTANCE( 1 ) ),WTR=T03XWTR,END
```

Where *xx* IIT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Modifying a Trace Instance to Disconnect an External Writer

In this example an existing trace instance is disconnected from an external writer.

```
TRACE CT,ON,COMP=ACT  
RR xx,OPTIONS=( INSTANCE( 1 ) ),WTR=DISCONNECT,END
```

Where *xx* IIT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Stopping a Trace Instance and Disconnecting an External Writer

In this example an existing trace instance is to be stopped and disconnected from an external writer.

```
TRACE CT,ON,COMP=ACT  
RR xx,OPTIONS=( INSTANCE( 1 ) HALT ),WTR=DISCONNECT,END
```

Where *xx* IIT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR910I Trace shutdown successful Instance(01)
```

Stopping an External
Writer

In this example an external writer is stopped. The writer must be disconnected from a trace instance before it can be stopped.

```
TRACE CT,WTRSTOP=T03XWTR
```

Processing Trace Data

In order to process the data collected with the TRACE command, use the IPCS CTRACE command. It handles trace data that is in the MVS Component Trace Entry (CTE) format. By collecting trace data in this format, it is possible to use IPCS facilities to format the trace data. IPCS exit routines need to be written to locate trace records in dumps, filter trace records, and drive trace record formats.

Troubleshooting

This chapter contains information about identifying and resolving problems.

The following topics are discussed:

- [Diagnostic Procedures](#)
- [Accessing the Online Client Support System](#)
- [CA-TLC: Total License Care](#)
- [Contacting Technical Support](#)
- [Generating a Problem Report](#)
- [Product Versions and Maintenance](#)
- [Requesting Enhancements](#)

Diagnostic Procedures

The following sections detail the procedures you should follow if you have a problem with a Computer Associates software product.

Collecting Diagnostic Data

In the table below, use the left column to categorize the problem your site encountered. Then, follow the instructions in the corresponding right column to generate useful diagnostic data.

Type of Problem	Procedure
What to Do if the NetMaster Command Processor Abnormally Terminates	<p>Under normal circumstances the NetMaster Command Processor abnormal termination should not occur.</p> <p>In the unlikely situation that it does, problem determination and diagnostic information must be retained to enable Computer Associates engineers to troubleshoot the problem. To ensure the problem is resolved as quickly as possible, the following diagnostic materials (if available) must be forwarded to Computer Associates technical support:</p> <ul style="list-style-type: none"> ■ The T09TCMDS dump (written to a CICS dump data set) ■ System and CICS/TS message logs ■ An SVC dump of the CICS region <p>Note: If the problem can be reproduced, supply a TCPEEP trace of TCP traffic over the NetMaster Command Processor port leading up to the problem.</p>
EZASOKET or EZACICAL call failing with bad Erno field	<p>Refer to IBM's <i>IP CICS Socket's Guide</i> and check the listing for all messages. Save all system output.</p> <p>If the erno message itself does not suggest what diagnostic approach to take please follow the procedure Debugging Guidelines for a Socket Management Application.</p>
Installation	<p>Refer to the install steps in <i>Getting Started</i>. Save all output.</p>
Socket Management Error Message	<p>Review message in the <i>Unicenter NetMaster Socket Management for CICS Message Guide</i>. If the message itself does not suggest what diagnostic approach to take, follow the procedure provided in Debugging Guidelines for a Socket Management Application.</p>

Debugging Guidelines for a Socket Management Application

Throughout this section the term *EZASOKET* can represent either an EZASOKET or EZACICAL application.

Note: Customer's with EZASOKET or EZACICAL API failures can debug their application by gathering the following information **all at one time**. Please do not send CA support pieces run from separate sessions as the time stamps will not match up between components, making a valid analysis impossible.

Follow the steps below to provide CA customer support with sufficient information to diagnose your issue. References to other sections are interspersed when needed for further detail.

Preparation Steps

1. Refer to [Step 1: Debugging Guidelines Required Maintenance PTFs](#) for possible maintenance requirements needed to perform the described diagnostics.
2. **Unicenter TCPaccess customers only:** Follow instructions in [Step 2: Raise the Size of Unicenter TCPaccess's Internal IFS \(Below the Line\) Trace Table](#).
3. Contact the vendor of the failing EZASOKET API application for specific debugging procedures available to their product.
4. Bring up component trace address space first and then CICS/TS. The member T09TRACE in the T09SAMP data set is an example set of JCL for the Component trace subsystem address space. See the Trace section in the "Diagnostic Commands" chapter for detailed installation and startup information.
5. Unicenter TCPaccess customers only: Issue MVS operator command to the TCPaccess address space to list PC and CSA module addresses in the JES logs (if the TCPaccess jobname is TCPIP):

```
F TCPIP,MODULE ALL
```
6. Start TCPEEP trace display formatting by following instructions in [Step 6 Part I: Required TCPEEP Command Examples](#).
7. Start Socket Management.
8. Start the application and wait for the hang or failure to occur.

When the Problem (abend or hang) occurs:

9. **Unicenter TCPaccess customers only:** Issue MVS operator command to the TCPaccess address space to dump and format API control blocks (If the TCPaccess job name is TCPIP):

```
F TCPIP,TCP SNAP ALL
```
10. Follow instructions in [Step 10 Part I: Take SVC Dumps of ALL the Related API Application Address Spaces](#).

11. Send CA support **everything**:

- All the TCP/IP job output including JES logs
- All the API application job output including JES logs
- SVC dumps of the TCP/IP, CICS/TS, and optionally T09TRACE
T09TRACE is required if your application falls in the description in [Step 6 Part II: Hard to Re-create \(Long Running\) Traces](#)
- Make sure you include the DCB attributes of the original SVC dump.
- TCPEEP output
- **Unicenter TCPaccess customers only:** TCP SNAP ALL output from the TCPaccess address space

***Important!** Please do not send support pieces run from separate sessions as the time stamps do not match up between components and a valid analysis is simply impossible.*

Step 1: Debugging Guidelines Required Maintenance PTFs

Unicenter TCPaccess required PTFs:

- The exact same procedures are used to debug IUCV or HPNS applications at the 5.2 level or above.
- EZA trace support was introduced at the 5.2 level of TCPaccess with PTF TP08600. EZA trace support was introduced at the 5.3 level of TCPaccess with PTF TP08599.

Step 2: Raise the Size of Unicenter TCPaccess's Internal IFS (Below the Line) Trace Table

This keeps the IFS internal trace table from wrapping too quickly. One can change the startup parameters or dynamically reallocate Unicenter TCPaccess address space internal trace table using the TRACE operator command.

In the Unicenter TCPaccess startup JCL there is a CMND parameter representing a member name in the SYSPROC DD library. You can set the IFS trace table to one MB by placing the following command in the CMND PDS member:

```
MODIFY TRACE ON SIZE(256)
```

To use one-half megabyte or above the line storage for the IFS trace table, set `SIZE(128)` on the `MODIFY` command.

You can dynamically reallocate Unicenter TCPaccess address space internal trace table using the `TRACE` operator command. Refer to the `TRACE` section under `IJT` commands in the *Unicenter TCPaccess Communications Server System Management Guide*, which documents the `TRACE` operator command.

A sample command to reallocate a large IFS trace table:

```
MODIFY TRACE ON SIZE(256)
```

Step 6 Part I: Required TCPEEP Command Examples

There are issues where it may take a long time to re-create a problem. Under these circumstances, long running TCPEEP traces can consume extremely large amounts of the JES SPOOL. The Socket Management trace address space can buffer large amounts of data inside the trace address space when using the MVS console operator command: `TRACE CT`.

If this condition applies to your situation, proceed to [Step 6 Part II: Hard to Re-create \(Long Running\) Traces](#), otherwise use the following instructions.

You can use TCPEEP to trace data, commands and responses through EZASOKET and out onto the network.

Start TCPEEP with the following for a TCP problem (limiting data to 16 bytes for issues where we are unconcerned with the data):

```
TCPEEP TRCSSID(ACR) +
GROUPS((NETIF),(IUCV,'MAXIUCVDATA(16)') , +
        FORMAT(TCP,DATA(16),IUCVDATA(16),EZADATA(16))
```

If running an UDP application, use the UDP invocation option.

```
TCPEEP TRCSSID(ACR) +
GROUPS((NETIF),(IUCV,'MAXIUCVDATA(16)') , +
        (EZA,'MAXIUCVDATA(16)')) +
        FORMAT(UDP,DATA(16),IUCVDATA(16),EZADATA(16))
```

You can limit TCP or UDP trace output when you invoke TCPEEP with the PORT(), HOST(), and PROTOCOL() subparameters on the GROUPS NETIF filter parameter. Refer to the “Diagnostic Commands” chapter in the *Unicenter TCPAccess Communications Server System Management Guide* for full information on limiting parameters.

Most problem issues can be associated with a server port and an application address space. Data integrity is not a key issue most problem issues, so one can greatly limit TCPEEP job output by limiting the amount of data printed and collected on the associated TCPEEP data parameters

In the following example, TCPEEP will trace TCP data for host IP address 10.12.14.32 and for batch job CICSPROD and the first 32 bytes of EZASOKET, HPNS and TCP data for an issue without data integrity issues:

```
TCPEEP TRCSSID(ACTR) +  
      FORMAT(TCP,DATA(32),IUCVDATA(32),EZADATA(32)) +  
      GROUPS((NETIF,'PROTOCOL(TCP),HOST(10.12.14.32),MAXDATA(32)'), +  
            (IUCV,'USER(CICSPROD),MAXIUCVDATA(32)' +  
            (EZA,'MAXEZADATA(32)'))
```

In an atypical problem where a user truly needs to sort out a problem with data integrity, a server using port 2151 in address space named CICSPROD can use these TCPEEP parameters:

```
TCPEEP TRCSSID(ACTR) +  
      FORMAT(TCP,DATA(32000),IUCVDATA(32000),EZADATA(32000)) +  
      GROUPS((NETIF,'PROTOCOL(TCP),PORT(2151),MAXDATA(32000)'), +  
            (IUCV,'USER(CICSPROD),MAXIUCVDATA(32000)' +  
            (EZA,'MAXEZADATA(32000)'))
```

Important note: *With the data amount set very high such as 32000 above, a very large trace is generated in a short period. This high a data value should be avoided unless you are absolutely convinced that there is a data integrity problem. For the vast majority of problems a glance value of 16 is sufficient. A value of 16 causes one line of data to display.*

Step 6 Part II: Hard to Re-create (Long Running) Traces

There are issues where it may take a long time to re-create a problem. Under these circumstances, long running TCPEEP traces can consume extremely large amounts of the JES SPOOL. The Socket Management trace address space can buffer large amounts of data inside the trace address space when utilizing the MVS console operator command: TRACE CT

Note: It is very important to stop the trace address space from wrapping by dumping the trace address space as close to the point of failure as possible. A few minutes of time on a busy system could easily overlay the needed trace information by wrapping.

Issue the following MVS console operator command to turn on system CT tracing where the subsystem of the trace address space is ACTR:

```
/TRACE CT,ON,COMP=ACTR
```

After the previous command is entered, the operator is prompted by Socket Management trace to enter the appropriate tracing related variables. Enter the GROUPS and BUFFERS related parameters in the OPTIONS field.

The GROUPS parameters are the exact same format as the GROUPS parameters in a TCPEEP command.

The BUFFERS(SIZE,NUMBER) variable in the trace address space allocates storage for the tracing. Setting BUFFERS to a large SIZE and NUMBER will reserve more storage for tracing.

- SIZE is in kilobytes and ranges from 64 to 1024
- NUMBER ranges from 2 to 128

In the following example, the trace address space gathers TCP data for port 3666; HPNS and EZA sockets API tracing for user ID CICSPROD; and the first 16 bytes of HPNS and EZA sockets API data :

```
R 82,OPTIONS=(GROUPS((NETIF,'PROTOCOL(TCP),PORT(3666),
MDATA(16)'),(IUCV,'USER(CICSPROD),MAXIUCVDATA(16)'),
(EZA,'USER(CICSPROD),MAXEZADATA(16)'),
BUFFERS(256,64)),END
```

At the point, when the failure occurs you must take an SVCDUMP of the trace address space to gather the trace data *before* the trace internally wraps. Refer to the SVCDUMP command discussion in [Step 10 Part I: Take SVC Dumps of ALL the Related API Application Address Spaces](#) to learn how to take an SVCDUMP of more than one address space.

The following command under IPCS for the SVC dump output can format the data using the GROUP and FORMAT parameters (the same exact parameter formats as those used with TCPEEP) are placed in the OPTIONS field where the subsystem of the trace address space is ACTR:

```
CTRACE SUMMARY COMP(ACTR) OPTIONS((FORMAT(TCP,DATA(16),IUCVDATA(16),EZADATA(16)),
GROUPS((NETIF,'PROTOCOL(TCP),MAXDATA(16)'),(IUCV,'MAXIUCVDATA(16)'),(EZA,'MAXEZAD
ATA(16)'))))
```

The IPCS CTRACE TALLY command gives you totals for each tracing record type when the tracing subsystemid is ACTR:

```
CTRACE TALLY COMP(ACTR)
```

Step 10 Part I: Take SVC Dumps of ALL the Related API Application Address Spaces

CA support needs SVC dumps of **ALL** the related API application address spaces including the CICS/TS region; the TCP/IP address space; and optionally the trace component address space.

Note: If you meet the criteria in [Step 6 Part II: Hard to Re-create \(Long Running\) Traces](#), then it is required that you also SVC dump the trace component address space as described below.

Refer to IBM manual *MVS System Commands* as the definitive source for both the SLIP and DUMP MVS operator commands.

In an **abend situation**, use the instructions in [Step 10 Part II: SLIP Command for an Abend Situation](#)

In a **hung application situation**, use the instructions in one of the following sections depending on the stack your are using (CA or IBM TCP/IP):

- [Step 10 Part III: \(Unicenter TCPaccess Only\) SVCDUMP Command for a Hung Application Situation](#) command for a hung application situation
- [Step 10 Part IV: \(IBM TCP/IP only\) SVCDUMP Command for a Hung Application Situation](#) for a hung application situation

Step 10 Part II: SLIP Command for an Abend Situation

The JOBLIST or ASIDLST parameters on the SLIP MVS operator command enables you to take a SVCDUMP of multiple address spaces when a single address space is going to abend. In our example below, we use the JOBLIST (or JL) parameter to dump all the API related address spaces:

- CICSPROD
- Trace address space (jobname T09TRACE)
- The TCP/IP stack (jobname of TCPIP)
- Each abend situation is different, set your slip trap for the abending address space at your site

In our example, address space CICSPROD abends with a S0C1 so the JOBNAME (abbreviated J) parameter must be set to CICSPROD.

Make sure we get CSA, all the private region, system trace, registers, RTM2WA, SQA, and LPA. This is done by setting SDATA parameters to:

```
CSA LPA LSQA PSA RGN SQA SUM TRT
```

You can use the following SLIP command to SVC dump both the CICSPROD, trace, and TCP/IP address spaces when CICSPROD takes a S0C1 abend:

```
SLIP SET,EN,A=SVCD,C=0C1,JL=(CICSPROD,TCPIP,T09TRACE),  
J=CICSPROD,SD=(CSA,LPA,LSQA,PSA,RGN,SQA,SUM,TRT),END
```

Important! Make sure that you can tell support the LRECL, RECFM, and BLKSIZE DCB attributes of the original SVC dump.

Step 10 Part III: (Unicenter TCPAccess Only) SVCDUMP Command for a Hung Application Situation

Unicenter TCPAccess's SVCDUMP operator command enables a user to dump multiple address spaces when a hang occurs.

You can dump multiple address spaces by specifying either:

- JOBNAME(<jobname1>,<jobname2>,...)
- ASID(<asid1>,<asid2>,...) parameters.

If you needed to dump three address spaces:

- TCPAccess (job name TCPIP)
- Trace address space (job name T09TRACE)
- Job name CICSPROD

Issue the following operator command:

```
F TCPIP,SVCDUMP JOBNAME(CICSPROD,T09TRACE)
```

Important! Make sure that you can tell support the LRECL, RECFM, and BLKSIZE DCB attributes of the original SVC dump.

Step 10 Part IV: (IBM TCP/IP only) SVCDUMP Command for a Hung Application Situation

If you are using standard MVS commands, follow the sequence below.

Important! Please note that matching the SDATA below exactly is critical to Computer Associates Support having adequate information to help you with your issue.

If you need to dump three address spaces:

- TCP/IP (job name TCPIP)
- Trace address space (job name T09TRACE)
- Job name CICSPROD

Use the following sequence of operator commands:

Note: *Initialized data* is site-dependent.

1. Enter: **DUMP COMM=(my socket app name hang in CICS)**
2. System response:


```
*98 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```
3. Enter: **R 98,JOBNAME=(CICSPROD,TCPIP,T09TRACE),CONT**

4. System response:

```
IEE600I REPLY TO 98 IS;JOBNAME=(CICSPROD,TCPIP,T09TRACE),CONT  
*99 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

5. Enter: **R 99,SDATA=(CSA,LPA,LSQA,PSA,RGN,SQA,SUM,TRT),END**

6. System response:

```
IEE600I REPLY TO 99  
IS;SDATA=(CSA,LPA,LSQA,PSA,RGN,SQA,SUM,TRT),END  
IEA794I SVC DUMP HAS CAPTURED: 237  
DUMPID=126 REQUESTED BY JOB (*MASTER*)  
DUMP TITLE=my socket app name hang in CICS
```

Important! *It is very important that you tell us the LRECL, RECFM, and BLKSIZE DCB attributes of the original SVC dump.*

Interpreting Diagnostic Data

Once you have collected the specified diagnostic data, write down your answers to the following questions:

1. What was the sequence of events prior to the error condition?
2. What circumstances existed when the problem occurred and what action did you take?
3. Has this situation occurred before? What was different then?
4. Did the problem occur after a particular PTF was applied or after a new release of the software was installed?
5. Have you recently installed a new release of the operating system?
6. Has the hardware configuration (tape drives, disk drives, and so forth) changed?

From your response to these questions and the diagnostic data, try to identify the cause and resolve the problem.

Accessing the Online Client Support System

Computer Associates is making extensive use of the Internet for your benefit. CA encourages you to *surf the net* to the CA home page at *ca.com* and the support site at *eSupport.ca.com*. The CA Internet site provides a great variety of information about CA products and services, including:

- Service and support
- Product information and sales
- CA-World conference information
- Press releases
- CA user groups

StarTCC, the web-based portion of CA-TCC (CA-Total Client Care) gives you real time, interactive access to CA product support information through the Internet.

Using StarTCC, you can:

- Open new issues
- Browse or update your existing issues and enhancement requests
- Perform keyword searches
- Download solutions, PTFs, and important notices regarding CA products, maintenance, and documentation

Requirements for Using StarTCC

The following are the requirements for using StarTCC:

- You must be a CA client with a current maintenance agreement.
- You must register through the CA Internet site.
- You must access the Internet with a browser that supports the HTML specification 2.0 or higher, such as Netscape Navigator 2.0 or higher or Microsoft Internet Explorer 3.0 or higher.
- Browsers that meet the HTML requirement support the following functions, which are required for StarTCC:
 - Secure sockets layer (SSL) to encrypt your transaction traffic
 - Encrypted data records (known as COOKIES)
 - HTML tables

StarTCC Security

StarTCC runs as a secured server (SSL). You may need to configure your browser to enable SSL. Guidelines for doing this are provided on the CA Technical Support page.

Accessing StarTCC

To access StarTCC, go to eSupport.ca.com. The StarTCC options are:

- StarTCC Information
- StarTCC Registration
- Access StarTCC

These options are described below.

StarTCC Information: Select the information option to view background information for StarTCC, details about the prerequisites, and instructions for configuring your browser. Be sure to review this section for updates or information not included here.

StarTCC Registration: Select the registration option to identify yourself to StarTCC. You must register before you can access StarTCC online. There are prompts for all required information, including your name, site ID, CA-StarTrak PIN, company name, E-Mail address, postal address, and desired password for accessing StarTCC.

Note: If you do not have a CA-StarTrak PIN, StarTCC provides one for you when you register.

Access StarTCC: Select the access option to begin using StarTCC. When prompted, enter your user ID and password. Once your sign-on is validated, you can perform the following:

- Open a new issue
- Open an issue for, or request an enhancement to, one of your CA products.
- Browse your issues and enhancement requests
- Display all issues for your site. The issues are grouped into three categories:
Open, Closed, and Enhancement Requests (DARs).
Browse and/or download solutions
- Specify criteria for selecting solutions, which you can then view or download.
- Search the CA knowledge base

- Specify criteria for searching the CA database for solutions, problems, and keywords that can provide you with immediate answers to your product support questions and concerns.
- Update your StarTCC profile
- Make changes to your default E-mail address, phone number, and password whenever necessary.
- Display your site's licenses
- View a list of all the CA products for which your company site is currently licensed.
- Display StarTCC news items
- View and download recently published solutions for CA products, instructions for downloading from StarTCC, and helpful information for using CA-StarTrak, StarTCC, or other CA products.

Accessing the Technical Support Phone Services Directory

The Computer Associates Technical Support Phone Services Directory lists each CA product and the telephone number to call for primary support for that product. To access the Support Phone Services Directory, set your browser for eSupport.ca.com and click on Contact Us.

CA-TLC: Total License Care

Many CA software solutions use license keys or authorization codes to validate your hardware configuration. If you need assistance obtaining a license key or authorization code, contact the CA-TLC: Total License Care group through eSupport.ca.com.

Contacting Technical Support

For further technical assistance with this product, please contact Computer Associates Technical Support at eSupport.ca.com for a complete list of CA locations and phone numbers. Technical Support is available 24 hours a day, seven days a week.

If you are unable to resolve the problem, please have the following information ready before contacting Computer Associates Technical Support:

- All the diagnostic information described in [Collecting Diagnostic Data](#).
- Product name, version number, operating system and genlevel.
- Product name and version number of any other software you suspect is involved.
- Version level and PUTLEVEL of the operating system.
- Your name, telephone number and extension (if any).
- Your company name.
- Your site ID.
- A severity code. This is a number (from one to four) that you assign to the problem.

Use the following to determine the severity of the problem:

1. A *system down* or inoperative condition.
2. A suspected high-impact condition associated with the product.
3. A question concerning product performance or an intermittent low-impact condition associated with the product.
4. A question concerning general product utilization or implementation.

Generating a Problem Report

Once a Computer Associates Technical Support representative has determined that your problem requires further investigation, use the CAISERV utility to generate a problem report.

CAISERV Utility

The CAISERV diagnostic facility produces a problem report for you to fill out and send in with all problem documentation. CAISERV also produces a short report on the Computer Associates products that you have installed. You should also send this information to help Technical Support solve your problem.

To invoke CAISERV, execute the CAISERV proc in your sample JCL library:

```
// EXEC CAISERV,CAILIB='CAI.CAILIB',  
// CAILPA='CAI.CAILPA',  
// CAICICS='NULLFILE',  
// SYSOUT=A  
//
```

Edit the JCL to your installation's standards, and submit the job.

The messages you may encounter when running CAISERV are:

CAPP999E INSUFFICIENT STORAGE TO PROCESS CAISERV

Explanation: Sufficient storage was not allocated to execute CAISERV.

User Response: Use at least 100 KB of storage for executing CAISERV.

Product Versions and Maintenance

New users of Unicenter NetMaster Socket Management are provided with a distribution tape containing the current version of the system. Clients are requested to operate only under currently supported versions of Socket Management.

Clients with current maintenance agreements also receive ongoing Socket Management maintenance. When a new version of the system is available, a notice is sent to all current Socket Management clients.

Requesting Enhancements

Computer Associates welcomes your suggestions for product enhancements. All suggestions are considered and acknowledged. You can use either of two methods to request enhancements:

- Contact your Account Manager who will initiate a Demand Analysis Request (DAR) for you.
- Enter your request through StarTCC, the CA web-based, interactive support system at eSupport.ca.com.

Security Program

This chapter provides information about the Unicenter NetMaster Socket Management for CICS Security Program. It describes how to invoke and use this optional security program feature provided with the LISTEN service.

About Optional Security Program

Security is provided via a program for user evaluation of requests via IP address or user ID/password for the services of local listeners/servers. If a security program is implemented, the user program is invoked for each connection request. The user program can be specified for each listener if desired. The appropriate server transaction is initiated if authorized by the user security program. Otherwise, the client is notified that the connection is terminated.

To implement the security program, the `SCTYEXIT=program-name` must be coded in the T09MCICS macro or the T09MLSTN macro of the T09CONez configuration table. This user program is EXEC CICS LINKed during the connection process, must conform to CICS coding standards, and be defined as a RDO program entry.

Note:

- If no SCTYEXIT parameter is coded in the configuration table, all connection requests are authorized and the user ID will be the same as the Listener transaction
- If SCTYEXIT is coded but the program is missing or is disabled, no connections are allowed

Each Listen tool can specify its own security program. If the Listen tool does not specify a security program and the SCTYEXIT parameter is coded on the T09MCICS configuration macro, then that program is used as the security program.

In other client/server designs, the application receives control when the connection is made and should make any desired security checks before starting server activity. However, by coding SCTYTYPE=MANDTORY and SCTYEXIT=program-name on the T09MCICS configuration macro, the security program name that is coded will always be executed regardless of what is coded at the T09MLSTN(listen tool) level or at the user-written listener before returning control to the application.

Security Program Logic Flow

The user security program is responsible not only for making the determination of whether a connection is authorized, but also for any desired logging or other capture of unauthorized requests. Because the program will be driven for each connection on a listener, performance implications should be considered in designing security programs.

When security is specified in the configuration table, a new transaction is started (the program is T09TLST2 with transaction ID IPT2). This transaction then CICS link to the specified security program. The program is passed the Security Communications Block (SCB). It contains fields used to determine the validity of the connection. One of the fields in the SCB is the token or socket ID of the connection. The token can be used to initiate SEND and RECEIVE calls in order to communicate with the remote client to determine a user ID, password, or any other identifying characteristics. Any of the other fields in the SCB can be used.

Security Program Return

Upon return from the security program, four SCB fields are used:

- The authorization switch is used to authorize the connection by setting a character one in the field.
- The terminal facility is used to specify a CICS termid to associate with the new CICS/TS transaction to be EXEC CICS STARTed.
- If the user ID field is specified the new transaction is STARTed with that user ID.
- The transaction to be started can also be modified by the security program. If changed then the newly specified transaction is STARTed.

Note: When termid and user ID are both specified, user ID takes precedence.

The security program can perform additional SEND and RECEIVE calls to request and retrieve data. This data might be some form of user ID or password. The program can then verify the user ID and password with the EXEC CICS VERIFY command. If the user ID is returned in the SCB, the new transaction is started with EXEC CICS START USERID (user ID).

The Security Communications Block

The connection process transaction and the user security program communicate through the SCB. Socket Management provides information about the request and its origin. The user security program determines whether the request is to be authorized and, optionally the name of a terminal facility or user ID to associate with an EXEC CICS STARTed server transaction. A DSECT of the SCB for assembler programs can be generated with the T09DSCTY macro.

This is what the T09DSCTY DSECT control block looks like in assembler language:

```
Name Operation Operands Description
SECPARM DSECT
SECTRAN DS CL4 SERVER TRANSACTION REQUESTED
SECDATA DS XL40 REQUESTOR DATA
SECSTRT DS CL2 HOW TASK IS TO BE STARTED
SECICTM DS XL6 INTERVAL CONTROL TIME
SECADRS DS 0CL8 REQUESTOR ADDRESS
SECAFAM DS H DOMAIN
SECRPRT DS H PORT
SECRHST DS F HOST IP ADDRESS
SECACTN DS CL1 PERMIT/PROHIBIT SWITCH
SECPMT EQU C'1' ..OKAY, INITIATE TASK
          DS X RESERVED
SECTMID DS CL4 ANY ASSOCIATED CICS TERMINAL
SECLPRT DS H LOCAL SERVER PORT
SECUSER DS CL8 USER ID
          DS CL512 RESERVED
SECTOKN DS F TOKEN - ENDPOINT
SECLHST DS F LOCAL HOST
*
SECLEN EQU *-&LABEL LENGTH OF SECURITY DATA AREA
```

The following table describes the data contained in the DSECT.

Field	Format	Description
SECTRAN	4-byte character	Requested server transaction, may be modified by the program.
SECDATA	40-byte character	Client data, if available.
SECSTRT	2-byte character	Method of server initiation: KC, TC, or IC.
SECICTM	6-byte character	IC Hours, Minutes, Seconds.
SECAFAM	2-byte binary	Address family: Inet domain=2.
SECRPRT	2-byte binary	Client remote port number.
SECRHST	4-byte binary	Client remote host IP address.

Field	Format	Description
SECACTN	1-byte character	Authorization switch: 1=accept 0=fail
SECTMID	4-byte character	Associated terminal facility.
SECLPRT	2-byte binary	Requested server local port.
SECUSER	8-byte binary	Returned user ID.
SECTOKN	4-byte binary	Token that represents the TCP connection.
SECLHST	4-byte binary	Local host IP address.

Unicenter NetMaster Administrator Interface

This appendix introduces you to the Unicenter NetMaster Socket Management Administrator Interface panels.

Starting the Interface

You can access the Unicenter NetMaster Administrator Interface via a menu shortcut. Type **/IPMON** on the command line and press Enter. Then type **?** in front of the CICS resource you want to view or for which you want to perform a task.

Moving Through the Panels

Use the commands on this panel to view and perform the task with which they are associated.

Command	Description
AL	View Alerts for a Resource
AM	Activate Monitoring
CL	Connections List via SocketMgmt
CMD	Command Entry - SocketMgmt
D	Display Resource Status
DUI	Delete SocketMgmt User Info. for Server
H	Show Performance History
IC	IP Connections
ICC	IP CICS Connections
IM	Inactivate Monitoring
SB	SocketMgmt and CPT Bounce
SQ	SocketMgmt Query Display
SS	CICS Server Start
SSB	SocketMgmt CMD Server Bounce
TS	CICS Transaction Start
UM	Update Resource Monitoring Definition
W	Display Outstanding WTORs for JOB/STC

Socket Management Query Display (SQ)

Use the SQ command to display a panel that enables you to view global information for the Socket Management product in the CICS address space.

```

DENM15----- SocketMgmt : Information -----Line 1 to 19 of 19
Command ==>>                               Scroll ==>> PAGE

SocketMgmt and CPT Summary
CICS Jobname ..... QATS22R1
TCP/IP Jobname ..... QA4QQ53
  SSID ..... AC4Q
  Trace SSID ..... TR4Q
CMD Server Address ..... 141.202.198.146
CMD Server Port ..... 2257

Running Products ..... +SOCKETMGMT
SocketMgmt Release ..... 1.0.0
CPT Release ..... 6.0.0
Startup Config Member .... T09CONFG
Security Exit Name ..... -

Connection Summary
Bytes Received ..... 94544
Bytes Sent ..... 409879
Calls Received ..... 2512
Calls Sent ..... 2442
***** BOTTOM OF DATA *****

F1=Help      F2=Split    F3=Exit      F5=Find      F6=Refresh
F7=Backward  F8=Forward  F9=Swap
    
```

Socket Management and CPT Bounce (SB)

Use the SB command to shut down and restart Socket Management in the CICS address space,

```

DENM15----- SocketMgmt : Bounce Confirmation -----
Command ==>>                               Function=Confirm

.- SocketMgmt and CPT Bounce Command -----
|
| WARNING: This command will stop and restart Socket Management and CPT
|
|   T09CONxx Startup Config Member Suffix      (default is current suffix)
|
| SocketMgmt and CPT Summary:
|   CICS Jobname ..... QATS22R1
|   TCP/IP Jobname ..... QA4QQ53
|   Running Products ..... +SOCKETMGMT
|   Startup Config Member ..... T09CONFG
|
| Press F6 to Confirm or F12 to Cancel
|
-----

F1=Help      F2=Split    F9=Swap      F6=Confirm
F7=Backward  F8=Forward  F12=Cancel
    
```

Socket Management CMD Server Bounce (SSB)

Use the SSB command to display a panel to stop and restart Socket Management in the CICS address space.

```

DENM15----- SocketMgmt : Bounce Confirmation -----
Command ==>                                         Function=Confirm

.- SocketMgmt and CPT Bounce Command -----
WARNING: This command will stop and restart Socket Management and CPT
        T09CONxx Startup Config Member Suffix      (default is current suffix)

SocketMgmt and CPT Summary:
CICS Jobname ..... QATS22R1
TCP/IP Jobname ..... QA4QQ53
Running Products ..... +SOCKETMGMT
Startup Config Member ..... T09CONFG

Press F6 to Confirm or F12 to Cancel

F1=Help      F2=Split      F9=Swap      F6=Confirm
                                           F12=Cancel

```

CICS Server Start (SS)

Use the SS command to start and EZASOKET server in the CICS address space.

```

DENM15----- SocketMgmt : Server Start Confirmation -----
Command ==>                                         Function=Confirm

.- Server Start Command -----
This command will start the server in CICS Jobname QATS22R1

Port .....
Transaction ID .....
Server Type ..... (CPT or EZA)
User ID .....

Press F6 to Confirm or F12 to Cancel

```

CICS Transaction Start (TS)

Use the TS command to start a transaction in the CICS address space.

```

DENM15----- SocketMgmt : Transaction Start Confirmation -----
Command ==>                                     Function=Confirm

.- Transaction Start Command -----
|
| This command will start the transaction in CICS Jobname QATS22R1
|
|   Tran
|   Parm
|
| Press F6 to Confirm or F12 to Cancel
|
|-----|

F1=Help      F2=Split      F9=Swap
F6=Confirm   F12=Cancel
    
```

IP CICS Connections (ICC)

Use the ICC command to display session information.

```

DENM15----- TCP/IP : CICS Socket Connections -----Stack: QA4QQ53
Command ==>                                     Scroll ==> PAGE

Line 1 of 3                                     Refresh Every ... Seconds
          DC=Display CICS P=Ping T=TraceRoute NL=Lookup S=View Z=Drop
          ZC=Drop CICS I=Information TPA=Transaction Path Analyzer L=Log ?=Actions
          Remote Local
Foreign Host  Port  Port User ID  TaskName  CICS Transaction  CICS
141.202.198.146 4106 1846 CICSUSER QATS22R1 IPTL 35 -
141.202.198.146 1846 4106 CICSUSER QATS22R1 QSC1 1184 -
141.202.200.71 6076 2257 - QATS22R1 IPCP 36 -
**END**

F1=Help      F2=Split      F3=Exit      F6=Refresh
F7=Backward  F8=Forward    F9=Swap      F11=Right
    
```

Connections List (CL)

Page 1

Use the CL command to display active connection information in the CICS address space.

```

DENM15----- SocketMgmt : Connections List -----
Command ==>                                     Scroll ==> PAGE

      CICS Jobname ..... QATS22R1
                                S/DC=Display Connection ZC=Drop Connection

SMID      Local  Remote  Remote      Connection  CICS      User ID
   3       1846   *      *           LISTENER    -         CICSUSER
   5       2257   *      *           LISTENER    -         CICSUSER
  2622    2257   6076   141.202.200.71 INBOUND     -         CICSUSER
  3559    2257   6093   141.202.200.71 INBOUND     -         CICSUSER
  3565    4107   1846   141.202.198.146 OUTBOUND    -         CICSUSER
  3567    1846   4107   141.202.198.146 INBOUND     -         CICSUSER
**END**

F1=Help      F2=Split    F3=Exit      F4=Return    F6=Refresh
F7=Backward  F8=Forward   F9=Swap      F11=Right

```

Note: You can select a specific session to view more information.

Command List (CL)
Page 2

```

DENM15----- SocketMgmt : Connections List -----
Command ==>                                     Scroll ==> PAGE

      CICS Jobname ..... QATS22R1
      S/DC=Display Connection ZC=Drop Connection

      Init  Init   Curr  Curr
      Tran Task   Tran Task
SMID  Name  Number Name Number  Session Type  VTAM LU
3     IPTL  35    IPTL  35    EZASOCKET      -
5     IPCP  36    IPCP  36    EZASOCKET      -
2622 IPCP  36    IPCP  36    EZASOCKET      -
3559 IPCP  36    IPCP  36    EZASOCKET      -
3565 QSC1  1290  QSC1  1290  EZASOCKET      -
3567 IPTL  35    QSS2  1295  EZASOCKET      -
**END**

F1=Help      F2=Split    F3=Exit     F4=Return   F6=Refresh
F7=Backward  F8=Forward  F9=Swap     F10=Left    F11=Right
    
```

Command List (CL)
Page 3

```

DENM15----- SocketMgmt : Connections List -----
Command ==>                                     Scroll ==> PAGE

      CICS Jobname ..... QATS22R1
      S/DC=Display Connection ZC=Drop Connection
      ----- Verb Statistics -----
      Response Times (ms)
SMID  Last Verb           In Progress  Last   Average  Calls  Errors  Time
3     * SRVR SELECT          In Progress  37259.17  12
5     SRVR ACCEPT          0.32        83.45    85      0      14:39
2622 SESS SELECT          7136.28    28576.01  341     0      14:39
3559 SESS READ            0.09        0.08     4       0      14:39
3565 SESS RECVFROM        0.06        61.49    2       0      14:39
3567 * SESS RECVFROM          In Progress  1104.23  2       0      14:39
**END**

F1=Help      F2=Split    F3=Exit     F4=Return   F6=Refresh
F7=Backward  F8=Forward  F9=Swap     F10=Left
    
```

Socket Management Initialization and Customization Services Parameter (admin function)

```

DENM15----- ICS : Initialization Parameters -----Page 1 of 1
Command ==>>>                                     Function=Browse

.- SOCKETMGMT - SocketMgmt Agents -----
|
| Dynamically Add Resources ..... YES (Yes or No)
|
| For NetMaster Background Region Access to CICS:
|   CICS User ID ..... CICSUSER
|   CICS User Password ..                               (This value is not displayed)
|
| For User Access to CICS:
|   Use NetMaster Signon Details ... YES (Yes or No; if No then the user ID
|                                         and password will be solicited
|                                         when the command is issued)
|
| Current ESP Receiver Status is ACTIVE
|
-----

.- Notes -----
| Dynamically discovered resources are added into the currently loaded
| System Image.
|
-----

F1=Help      F2=Split      F3=Exit      F4=Update      F5=ILog
    
```


The Terminal Administrator Interface

This appendix describes the CICS/TS transactions and panels of the Unicenter NetMaster Socket Management Administrator Interface.

It discusses the following topics:

- [Utilization](#) – Describes monitoring, tracing, and termination of TCP connections
- [Configuration](#) Table Parameters panel – Describes how to display, update, and add Socket Management Tool definitions
- [Operations](#) – Describes startup and graceful termination of the Socket Management Interface, and control of the diagnostic tool and statistics

The Socket Management Administrator Interface is a set of CICS/TS transactions and screens that improve visibility and control of the TCP/IP connections being established through the Socket Management Interface. It provides a system administrator with the tools to trace, diagnose, and terminate any CICS/TS transaction using TCP or UDP calls.

The Administrator Interface also enables dynamic modification of the Socket Management tools defined in the Configuration Table, and the ability to create new Socket Management tool definitions.

Administrator Interface features are grouped into the following major components:

- Utilization:** Monitoring, tracing, and termination of TCP connections.
- Configuration:** Display, update, and addition of Configuration Table entries.
- Operations:** Start up and graceful termination of the Socket Management Interface, and control of diagnostic tools and statistics.

The following sections describe each of the CICS/TS screens (called panels) used by the Administrator Interface, and the services they provide to a system administrator.

Entering the Administrator Interface

The Main Menu for the Administrator Interface is invoked with the IPAM transaction. This transaction produces the Main Menu display as shown in this example:

Note: All the Administrator Interface transactions have a three-character prefix of IPA. You can change this default at installation if these transaction IDs conflict with existing CICS/TS *transids*. (The default IPAM transaction is given here.)

The Main Menu panel contains selections for the three main entry points into the Administrator Interface: Utilization, Configuration, and Operations. There is also a selection for Navigation that describes the standard PF key and function code assignments.

```

                                WELCOME TO THE
                                COMPUTER ASSOCIATES
                                Unicenter NetMaster/Unicenter TCPaccess
                                ADMINISTRATOR INTERFACE

                                PF4) U: UTILIZATION
                                PF5) C: CONFIGURATION
                                PF6) O: OPERATIONS
                                PF9) K: NAVIGATION
    
```

Function Codes and PF Keys

You can make a selection on the Main Menu:

- Using a PF key
- Entering a function code (a one- or two-character command typed on the command line at the bottom of most Administrator Interface panels)

For example, to go to the first Utilization panel from the Main Menu, you can either press the PF4, or type **U** on the command line and press Enter.

The command line in a panel is identified by the > symbol. The command line is a 12-byte input field, followed by an error message field on the same line.

Every panel in the Administrator Interface makes this combination of PF keys and function codes available to you for navigating from panel to panel. The PF key choices appear at the bottom of each panel to assist novice users, while the function codes are available for experienced users who want to jump directly to a specific panel.

Note: Function assignments for PF keys 13–24 are always equivalent to the assignment of PF keys 1–12.

Administrator Interface Navigation

The fourth selection, Navigation, on the Main Menu brings up the online explanation of the standard PF key values and navigation function codes.

The following is a sample panel.

```

ADMINISTRATOR INTERFACE NAVIGATION

At all times      CLEAR KEY   or 'X' =  ESCAPE to exit and return to CICS
                  PF1/PF13   or '?' =  HELP to learn about the current screen

Usually          PF3/PF15           =  goback to return to the parent screen
                  PF4/PF16   or 'U' =  Utilization to see tasks
                  PF5/PF17   or 'C' =  Configuration to see parameters

If appropriate   PF7/PF19   or 'B' =  Page Backward through a list
                  PF8/PF20 or 'F' =  Page Forward through a list

Each function in the Interface has an associated 1- or 2-byte function code.
Utilization, Configuration and Operations may be entered directly from CICS
by appending their 1-byte code (U/C/O) to your 3-byte system prefix, creating
a CICS transaction id.  In addition, a 1- or 2-byte function code may be
specified as a parameter to the Main Interface program or may be input while
in another function to jump over directly.  Function codes are found prefixing
the panel identifiers in the upper right hand corner of each display.

```

As this panel shows, almost every Administrator Interface panel has a brief help display that you invoke using the PF1 key or the ? function code.

The PF3 key exits a panel and returns you to the parent panel, the PF8 key browses forward through a series of panels, and the PF7 key browses backward through a series of panels.

Most of the Administrator Interface panels have a panel ID in the upper-left corner. The two-character prefix of the panel ID is the function code for the panel.

Note: You cannot jump to all panels directly. Some panels must have the displayed item selected directly or browsed from a previous panel.

Exiting the Administrator Interface

You can use the Clear key or X function code to exit the Administrator Interface from any panel. After exiting, the following message appears, confirming the exit.

ADMINISTRATOR INTERFACE TERMINATED

Bypassing the Main Menu

In order to enter the Administrator Interface faster, you can bypass the main menu by entering the transaction ID for each major component directly from a cleared screen. Once inside the Administrator Interface, these three major panels can always be jumped to by entering their function code or PF key.

Trans Code	Primary Panel Name for Component	Function Code	PF Key
IPAU	Utilization Summary panel	U	PF4
IPAC	Configuration Table panel	C	PF5
IPAO	Operations Control Menu	O	None

Note: It is recommended that only these three transaction IDs and IPAM be used to directly enter the Administrator Interface. Using any of the other Administrator Interface transactions with the IPA prefix can have unpredictable results.

Utilization

The Utilization component of the Administrator Interface provides a realtime display of all the CICS/TS tasks in the system that have a TCP connection or a UDP endpoint established.

Utilization panels can also display detail information about the current state of each connection endpoint and enable the administrator to perform the following types of actions on those connections:

- Terminate any active connection by making an abortive close on the connection
- Ping the remote host of a connection to make sure that the remote host is still alive
- Dynamically set and remove trace options on a connection for any Socket Management tool or API program, and observe the trace output in the Online Trace

These features give a CICS/TS administrator the ability to monitor and diagnose all of the CICS/TS transactions that are using TCP or UDP on that CICS/TS system.

Utilization Summary Panel

The Utilization Summary panel is the heart of the Administrator Interface. It displays all of the endpoints established through the Socket Management Interface.

Three types of Socket Management endpoints can be displayed:

- A TCP connection endpoint
- A UDP endpoint (which is connection-less)
- A listening endpoint (which is waiting for a connection)

Each of these endpoint types can be active and owned by a CICS/TS task, or inactive with no current task owning it.

The following is an example a Utilization Summary.

```

1 mm/dd/yyyy          UTILIZATION SUMMARY          2 US01
17:39:14 3          INTERFACE STARTED AT 17:20 ON 04/11/2002 IN APPLID QATS22R2

4RECV:      30 MSGS@          1097 BYTES.  6      TASKS:      6      TOT:      47
5SEND:      31 MSGS@          1909 BYTES.  7  STORAGE:    43K    HWM:     47K

      TOKEN      TASKNO  TRANID   PORT      REMOTE HOST:PORT      STATE
8  156BC008      34     IPTL   3021 TCP
  156C0008      35     IPTL   3022 TCP                SELECT
  157B9008      36     IPCP   3024 TCP                SELECT
9  157BD008      36     IPCP   3024 TCP 141.202.198.171:1126  ACCEPT
  157D5148      63     JC01   4098 TCP 141.202.198.145:3022  RECVFROM
  157C2BB8      69     JT01   3022 TCP 141.202.198.145:4098  RECVFROM

10 >          11...ENTER TOKEN OR TASKNO TO SEE DETAIL INFORMATION...
    Pf1=?   Pf3=goback   Pf4=refresh   Pf5=Configuration
12 Pf2=Inactives   Pf6=Gen Entry   Pf10=Browse Servers   Pf11=Browse Clients

```

Where:

- 1 Current date and time on this CICS/TS system.
- 2 Panel title and panel ID for this screen.
- 3 Date and time the Socket Management Interface was initialized, and the APPLID of this CICS/TS system.
- 4 Total number of Socket Management messages received by all Socket Management endpoints since Socket Management initialization, and their cumulative byte length.
- 5 Total number of Socket Management messages sent by all Socket Management endpoints since Socket Management initialization, and their cumulative byte length.
- 6 Current number of active Socket Management tasks in this system, and the total number of Socket Management tasks that have executed since Socket Management initialization.
- 7 Total amount of storage currently being used for Socket Management control blocks and buffers, and the high-water-mark for that storage since Socket Management initialization.
- 8 Inactive Socket Management endpoints that have no active CICS/TS task attached.
- 9 Active Socket Management endpoints that display the TASKNO of their attached CICS/TS task, and the host IP Address and port number of their remote partner.
- 10 Command line used for selecting the detail display of a particular connection, by entering its TASKNO or TOKEN address.

- 11 Message area for command feedback and error notification.
- 12 PF keys currently active for this panel.

Summary Statistics

The Summary Statistics shown at the top of the Utilization Summary panel are cumulative for all Socket Management transactions in this CICS/TS system.

- The total number of Socket Management messages sent and received since Socket Management initialization and their cumulative byte length are shown on the left
- The current number of active Socket Management tasks in the system and the total number of Socket Management tasks that have executed since Socket Management initialization are shown on the right
- Below the Socket Management messages and tasks is the total amount of storage currently used for Socket Management control block and buffers, and the high-water-mark (HWM) for that storage since Socket Management initialization

Note: If the Reset All Statistics function is performed from the Operations Control menu, the message and byte counts are reset to zero, and the task and storage counts are reset to their current values.

List of Socket Management Endpoints

Each page of the Utilization Summary can display 13 Socket Management endpoints. The endpoint for a TCP connection can exist across CICS/TS tasks and is passed via the GIVE and TAKE API calls.

If a CICS/TS task currently owns an endpoint, then the connection is active and the task number of the owning CICS/TS task is displayed in the TASKNO column. If the endpoint has no owning CICS/TS task at the moment, it is inactive and the TASKNO column is blank.

- Active endpoints – The screen line is highlighted; the task number is shown.
- Inactive endpoints – The line is not highlighted; no task number is shown.

The Socket Management endpoints are displayed in task number sequence, so the inactive endpoints are shown first, then the active endpoints are shown, with the oldest tasks first.

Endpoint Data Fields	<p>Each Socket Management endpoint listed on the Utilization Summary panel has these data fields displayed in each of its columns:</p> <p>TOKEN – The address of the control block that represents the Connection EndPoint.</p> <p>TASKNO – The task number of the owning CICS/TS task. This field is blank for inactive endpoints.</p> <p>TRANID – The transaction ID of the owning CICS/TS task.</p> <p>PORT – The local port number being used, and the type of protocol being used (TCP or UDP).</p> <p>REMOTE HOST:PORT – The IP address and port number of the partner program on the remote host. Listener tasks have blanks in this field.</p> <p>STATE – The STATE column contains the last API call made on this endpoint, and the return code of that call.</p>
Example Syntax	<p>API Call Return Code</p> <p><i>XXXX-RRR</i></p> <ul style="list-style-type: none">■ API Calls<ul style="list-style-type: none">TCP API Calls: CLOSE, CONNCT, GIVE, LISTEN, RECV, SEND, TAKE.UDP API Calls: RCFR, SNT0.■ Return Codes:<ul style="list-style-type: none">I/P – In progress. The call is still being processed by the TCP transport provider.OK – A zero return code was returned for the call.nnn – A three-digit, decimal return code, as specified in the <i>Unicenter NetMaster Socket Management for CICS Messages and Codes</i> or in a return code copy member such as T09KCRCS for COBOL.

Displaying Inactive Endpoints

To shorten the list of endpoints displayed on this panel, you can suppress the display of all inactive endpoints. The inactive endpoints are not currently owned by any CICS/TS task. The PF2 key toggles the display of these inactive endpoints on the Utilization Summary panel.

When inactive endpoints are displayed, the PF2 key is shown as:

```
PF2=Inactives off
```

When inactive endpoints are suppressed, the PF2 key is shown as:

```
PF2=Inactives
```

Note: The Inactives off option remains in effect for only one administrative session. When first displaying the Utilization Summary panel the default is always Inactives.

Explicit Selection of the Detail Display

A detail display can be selected for any of the endpoints displayed on the Utilization Summary panel.

To do this, type either the TASKNO or TOKEN field number on the command line, as the default prompt for this panel suggests:

```
...ENTER TOKEN OR TASKNO TO SEE DETAIL INFORMATION...
```

The task number is the easiest field to enter for this selection. However, the TOKEN address field must be offered as an alternative for the inactive endpoints that do not currently have an owning CICS/TS task, and thus no task number to enter.

If an invalid number is entered on the command line, then the following message appears, prompting for another TASKNO or TOKEN:

```
REQUESTED TASK NOT FOUND - PLEASE VERIFY          1234
```

Browsing Detail for Client or Server Task

These are the options for browsing through the detail displays if you do not want to select a task explicitly on the Utilization Summary panel:

PF10=Browse Servers – (Connection initiated by remote host)

PF11=Browse Clients – (Connection initiated by CICS/TS task)

A client task initiates a connection with a server on a remote host. A client task does not need to specify what local port it will be using, so the TCP provider automatically assigns its connection a port number, beginning with 4096 and going up.

A server task must always keep the same port number and should not conflict with any other server or client port. There may be a listener task and several subordinate receiver tasks started by the listener that are all associated with the same server port.

Generating an Online Trace Entry

You can specify online tracing options from the Browse Detail panel, or by using PF6 on the Utilization Summary panel, as shown by this option:

PF6=Gen Entry

Use this option when you want to trace an endpoint that is not currently established. The Browse Detail panel can only specify tracing options for connection endpoints that currently exist. However, some connections complete so quickly, that the Online Trace entry must be defined prior to the connection initiation in order to capture it. How that trace entry is specified, and how its trace options are selected, are described further in [Utilization](#).

Client/Server Browse Detail Panel

The Browse Detail panel displays statistics for a particular endpoint and its CICS/TS transaction. From this panel, three potential actions can be taken on the endpoint:

- End its CICS/TS task, and close the endpoint
- Ping the remote host for the endpoint to test if it is still alive
- Set API trace options for the endpoint

The Browse Detail panel can display or suppress different data fields, depending on the type of CICS/TS task and the state of the endpoint. Three examples follow that show these different types of Browse Detail displays.

Examples

This section presents three usage examples.

Example 1: Displaying an Active Connection Endpoint

The first example is a server transaction waiting to receive more input on a TCP connection with a remote client. The identification line in the center of the screen (item 6) displays the data fields that identify this endpoint:

- Task Number
- Local Port
- Token Address
- Protocol
- State (last API call, and Return Code)

Note: This identification is very similar to the line on the Utilization Summary for this endpoint. Refer to [Utilization](#) for detailed descriptions of these fields.

The First Activity and Last Activity fields show the date and time of the first API call made on this connection, and the data and time of the most recent API call made, which is shown in the State field along with its return code. In this case, the I/P in the State field indicates that the Receive call is still 'In Progress'. The Server task is waiting for its last call to receive more input on the connection.

Because this endpoint has an active connection with a remote client, all the remote host information is shown, and the Endpoint Statistics are shown.

The remote host information shows the IP address of the remote host in dotted decimal notation. Following the colon is the port number being used by the Client program on the remote host. If the Domain Name Resolver is active, then the domain name of the remote host will also be displayed.

Server Transaction
Detail

The following figure shows a page of the Browse Detail panels invoked by the Browse Servers function. It displays one of the two current server transactions receiving input on port 1804.

```

1 04/27/01 CLIENT/SERVER BROWSE DETAIL 2 BD01
18:34:15 4 SERVER TRANSID TFS2 ON PORT 1804 P. 3

+---CUMULATIVE STATISTICS FROM 04/27/01 AT 14:05:05 -----+
| RECEIVED MESSAGES: 1391 SENT MESSAGES: 1391 |
| BYTES: 51928 BYTES: 51928 |
5 |
| TOTAL CONNECTIONS: 3 CURRENT CONNECTIONS: 2 |
+-----+

6 TASK#: 34 LPORT: 1804 TOKEN: 00092B78 PROTOCOL: TCP STATE: RECV--I/P
7 FIRST ACTIVITY@: 04/27/01 14:05:13 LAST ACTIVITY@: 04/27/01 18:33:48
8 REMOTE ADDRESS: 138.70.53.17:4365
9 HOST NAME:
10 MESSAGES RECV'D: 675 SENT: 675
    BYTES RECV'D: 24738 SENT: 24738
(MORE 12)

11 >
13 Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration Pf7=Bwd Pf8=Fwd
    Pf2=End Task (close) Pf6=Ping Remote Pf9=Trace Options

```

Where:

- 1 Current date and time on this CICS/TS system.
- 2 Title of this panel.
- 3 Panel ID and current page number.
- 4 Transaction ID of the server task and the local port it is using.
- 5 Cumulative statistics being collected for all instances of the server transactions using this port number.
- 6 Identification line for this TCP endpoint, giving its task number, local port number, token address, protocol, and current state.
- 7 Date and time of the first API call on this connection, and the date and time of the most recent API call, which is shown in the state field.
- 8 IP address of the remote host in dotted decimal notation, followed by the port number being used by the client program on the remote host.
- 9 Domain name of the remote host as resolved by the DNR subsystem.
- 10 Endpoint statistics collected for just this specific connection.
- 11 Command line used for specifying an Administrator Interface function code.

12 (MORE) indicator that more panels are available as part of a browse.

13 PF keys currently active for this panel.

The counters for both the Cumulative Statistics and the Endpoint Statistics can be reset to zero from the Operations Menu in the Administrator Interface, as described in [Operations](#). The date and time shown in the Cumulative Statistics box is the last time the statistics were reset or when the Socket Management Interface was initialized.

Example: 2
Listener Transaction
Detail

This second example is a Socket Management listener task waiting to receive new connection requests on the same local port number as the previous server example.

The following figure shows a single Browse Detail panel that was selected directly by TASKNO from the Utilization Summary panel.

```

1 04/27/01 CLIENT/SERVER BROWSE DETAIL 2 BD01 3
18:36:11 4 SERVER TRANSID TPS4 ON PORT 1804

+---CUMULATIVE STATISTICS FROM 04/27/01 AT 14:05:05 -----+
| RECEIVED MESSAGES:          1391          SENT MESSAGES:          1391 |
| BYTES:                      51928        BYTES:                      51928 |
| TOTAL CONNECTIONS:           3           CURRENT CONNECTIONS:    2 |
+-----+

6 TASK#:          31 LPORT: 1804 TOKEN: 000922F8 PROTOCOL: TCP STATE: LISTEN-I/P

7 >
8 Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration
Pf2=End Task (stop listen) Pf9=Trace Options

```

Where:

- 1 Current date and time on this CICS/TS system.
- 2 Title of this panel.
- 3 Panel ID.
- 4 Transaction ID of the server task and the local port it is using.
- 5 Cumulative statistics being collected for all instances of the server transactions using this port number.

- 6 Identification line for this listen endpoint, giving its task number, local port number, token address, protocol, and current state.
- 7 Command line used for specifying an Administrator Interface function code.
- 8 PF keys currently active for this panel.

Note: This panel is missing the following data fields that were included in Example 1 (Browse Detail) :

- No remote host information is shown because a listener task has no established connection. It is just waiting for a new connection request to come in from any remote host.
- No endpoint statistics are shown because the listener task has only a listening endpoint, and no established connection.
- No PF6 key to ping the remote host since no connection exists.
- No PF7 or PF8 key since this panel is not part of a browse.
- No page number is shown since this panel is not part of a browse.

Example 3
Displaying a Listener
Endpoint

A listener task has an endpoint, but no connection with a remote host. It usually starts a separate server task when a new connection request comes in from a remote host. This is why the bottom half of the screen is empty for a listener task, but is filled in for its server task.

The Cumulative Statistics shown in this example are the same as those shown in the server task example, because the counts for connections and data transmissions are accumulated based on the local port number of the servers. The listener task and all of its subordinate server tasks (which it started) all have the same local port number, and will all show the same set of Cumulative Statistics.

Example: 4
Client UDP
Transaction Detail

This example is a UDP Client transaction that sends a datagram to the echo port of a remote host.

The following figure shows a page of the Browse Detail panels invoked by the Browse Clients function.

```

1 04/27/01          CLIENT/SERVER BROWSE DETAIL 2          BD0 3
18:39:48          4 CLIENT TRANSID U0S2 TO PORT 7          P. 1
5          5          STATISTICS CAPTURE CURRENTLY OFF

6 TASK#:          35 LPORT: 4365 TOKEN: 00098528 PROTOCOL: UDP STATE: RCFR--OK
FIRST ACTIVITY@: 04/27/01 14:08:21 LAST ACTIVITY@: 04/27/01 18:35:56
REMOTE ADDRESS: 138.70.45.12:7
HOST NAME:

MESSAGES RECV'D:          1          SENT:          1
BYTES RECV'D:          15          SENT:          15
(MORE 8
7 >
9 Pf1=? Pf3=goback Pf4=Utilization Pf5=Configuration Pf8=Fwd
Pf2=End Task (close) Pf6=Ping Remote Pf9=Trace Options

```

Displaying a UDP
Endpoint

A UDP task has an endpoint, but no connection with a remote host, whether it is a client or a server transaction.

Because a UDP client must specify a remote host and port number, that information can be displayed, along with the endpoint statistics for amount of data sent and received by the UDP task.

Note: The top half of the screen is empty where the Cumulative Statistics would appear. This is not because the Client is a UDP task, but because AISTATOP=NO was specified in the Socket Management Configuration Table for this execution of CICS/TS. When this parameter is set to NO, every Browse Detail panel displays the following message:

STATISTICS CAPTURE CURRENTLY OFF.

Terminating the Endpoint Task

Use the PF2 key to terminate the CICS/TS task for the endpoint displayed on the Browse Detail panel. This termination is done by an abortive close on the Socket Management endpoint, not by abending the CICS/TS task explicitly.

This action causes a disconnect return code to any outstanding API call from the CICS/TS task, which must then interpret that disconnect return code, and terminate via its own logic. When the PF2 key is pressed, the Quiesce/Confirm panel appears to confirm the close of the displayed task.

```

04/27/01                QUIESCE/CLOSE CONFIRMATION    ETO1
18:40:59

TASKNO      PORT      REMOTE HOST:PORT      TOKEN      TRANID
-----
      212      4394      138.70.45.12:0007    000C59B8    TPC2

PLEASE USE PF9 (QV) TO CONFIRM REQUEST TO QUIESCE THIS TASK!

>
Pf1=?   Pf3=goback   Pf4=Utilization   Pf5=Configuration

```

Confirming the Quiesce

The PF9 key confirms the termination, or the QV function code can be entered on the command line.

After confirmation, the next panel that appears is the Utilization Summary panel, to verify the status of the task.

The following response message appears in the message area of that panel:

```
> .....REQUESTED CONNECTION(S) CLOSED .....
```

If the termination is to be aborted by not confirming on the Quiesce/Confirm panel, any of the standard navigation function codes or PF keys can be used to go to another panel. The PF3 key returns to the Utilization Summary panel, not the previously displayed Browse Detail panel.

Pinging the Remote Host

Use the PF6 key to send a ping to the remote host displayed on the Browse Detail panel.

The ping is an ICMP echo request that does not use the displayed connection, but can test if the remote host is still reachable and if its system is still active. If a break has occurred in the route to the host, or if its system has crashed, the following message appears on the Browse Detail panel:

```
> ...NO RESPONSE FROM REMOTE HOST....
```

If the ping is successful, then the following message appears:

```
> ...REMOTE HOST IS ALIVE....
```

This feature can help in determining if the displayed endpoint needs to be terminated because the remote host is no longer reachable.

Configuration

The Configuration component of the Administrator Interface enables you to display and modify the Socket Management tool definitions made in the Socket Management configuration table. The Administrator Interface now lets you modify many of the definitions of the Socket Management tools in the following ways:

- Update the parameters of any Listen/Receive tool
- Add a new Listen/Receive tool to the configuration table
- Start any of the Listen/Receive tools defined to the configuration table

Some global parameters of the configuration table are also displayed by the Configuration component, but they cannot be modified. This is because they define the attributes of the currently running Socket Management Interface (such as the subsystem ID of its TCP transport provider), which cannot be modified without stopping and restarting the Socket Management Interface.

Note: The *dynamic updates* to the tool parameters, and the addition of new tool definitions, are *not permanent changes* to the configuration table. The next time the Socket Management Interface is initialized, the configuration tools revert to those definitions assembled into load module T09CONez, depending on the suffix identified.

Configuration Table Parameters Panel

The Configuration Table Parameters panel displays all of the global attributes of the Socket Management Interface that is currently running. These attributes are specified by the T09MCICS macro coded at the start of the configuration table. This panel shows both the default attributes, and those that were set by macro parameters for the current definition of the Socket Management Interface. These parameters cannot be modified dynamically.

The Configuration Table Parameters panel is invoked by the C function code from anywhere in the Administrator Interface, or by using the PF5 key, when this option is displayed at the bottom of a panel:

PF5=Configuration

The following is a sample of the Configuration Table Parameters panel.

```

1 04/27/01                CONFIGURATION TABLE PARAMETERS 2          CT01 3
18:54:32 4 CPT INTERFACE STARTED AT 14:04 ON 04/27/01 IN APPLID CICSQA01

5 LOCAL HOST NAME:  LOKI
  ADDRESS:  138.99.128.15

6 TCP/IP SUBSYSTEM:  TCP IP 7      DNR SUBSYSTEM:  8          CICS AMXT:  100

TRANSACTION NAMES:  TERMINATION = IPPR      LISTEN = IPTL
9 TD QUEUE NAMES:  STATISTICS = ACST      TRACE = ACTR      ERRORS = ACER
10 TRANSLATION TABLE:  T09XENG
15 IUCV:  LINGER = 1  MSOCK = 50  QLSTN = 5  MSGL = 10  MTAKE = 5

ADM INT STATISTICS TABLE:  8,000 BYTES  165 ENTRIES
STATISTICS DISCARD TDQ:  ACOS
11 MAX TRACE RECORDS:  800
MAX ERROR RECORDS:  500
COMPONENT NAME PREFIX:  IPA
OPTIONS:  CAPTURE CLIENT/SERVER STATISTICS

12 >
14 Pf1=?  Pf3=goback  Pf4=Utilization 13 Pf5=Configuration
Pf2=Listen Tools  Pf6=Send Tools
    
```

Where:

- 1 Current date and time on this CICS/TS system.
- 2 Title of this panel.
- 3 Panel ID.
- 4 The date and time the Socket Management Interface was initialized and the APPLID of this CICS/TS system.
- 5 The local host name and IP address of the TCP transport provider to which this Socket Management Interface is connected.
- 6 The subsystem ID by which the TCP transport is specified.

- 7 The subsystem ID by which the Domain Name Resolver for IP address/name translations. (Only for TCPAccess.)
- 8 The CICS/TS AMXT parameter from the SIT table specifying Active Max Tasks.
- 9 The transaction names and transient data queue names used by Socket Management.
- 10 The load module containing the ASCII/EBCDIC translation table.
- 11 The parameters for the Administrator Interface itself.
- 12 The command line used for entering a function code.
- 13 The message area for command feedback and error notification.
- 14 The PF keys currently active for this panel.
- 15 Only displays when running with IBM TCP/IP.

Configuration Table Description

Most of these attributes will rarely change for the Socket Management Interface. The following list provides a quick overview of the attributes before browsing the Socket Management tool definitions.

Local Host Name	Full domain name of the local host's TCP/IP subsystem.
Address	Dotted decimal IP address of the local host's TCP/IP subsystem.
TCP/IP Subsystem	Subsystem ID of the TCP/IP transport provider.
	Note: The local host name and address are not explicitly specified in the configuration table. They are derived from the Jobaname= parameter that specifies which subsystem will provide TCP/IP services for this CICS/TS.
DNR Subsystem	Subsystem ID of the Domain Name Resolver for IP Name/IP Address translation.
CICS AMXT	CICS/TS Active Max Task value from the CICS/TS system initialization table.
	Note: This CICS/TS parameter is shown here because Listener tasks are long running tasks. Creating and starting new Listener tools could adversely impact a CICS/TS system with a low AMXT value.
Transaction Names	The configuration table lets you change these transids, if the default names are already in use by this CICS/TS.
TD Queue Names	The configuration table lets you change these queue names, if the default names are already in use by this CICS/TS.

Translation Table	<p>The configuration table lets you specify a custom ASCII/EBCDIC translation table, if the default table provided is not sufficient.</p> <p>Note: The Configuration Table Parameters panel shows the default values for all the transaction names, queue names, and translation table.</p>
Linger	<p>Shows the default number of seconds to wait after an orderly close call.</p>
MSOCK	<p>Shows the default maximum number of sockets allowed for the IUCV path of each Listener task on a port.</p>
QLSTN	<p>Default number of connection requests that can be queued in a backlog for each Listener task on a port.</p>
MSGL	<p>Maximum number of IUCV messages allowed at any one time.</p>
MTAKE	<p>Maximum number of seconds to wait for a TAKE call to be issued for a token. The task started by the Listener should issue this TAKE call after it retrieves the token passed to it by the Listener.</p>
Adm Int Statistics Table	<p>Number of bytes and the number of entries allocated to the internal statistics table used by the Administrator Interface.</p> <p>Default: 1000 bytes.</p> <p>Note: The Configuration Table Parameters panel shows the values when parameter AITABSZ=8000 is specified in the configuration table.</p>
Statistics Discard Queue	<p>When the Cumulative Statistics are reset, or if Socket Management is shut down, the entries in the statistics table are written as data records to this transient data queue.</p> <p>Default: No queue defined.</p> <p>Note: If the queue name specified in this field is not defined in the CICS/TS RDO, then the following error message appears:</p> <pre>TRANSIENT DATA QUEUE XXXX IS NOT DEFINED TO CICS</pre>
Max Trace Records	<p>Maximum trace entries that can be collected by the Online Trace. You can reset this value from the Operations Control menu.</p> <p>Default: Zero.</p>
Max Error Records	<p>Maximum Error Log record entries that can be displayed from the Operations Control menu.</p> <p>Default: 30.</p>

Component Name Prefix	Display the three-character prefix of all the transids used for the Administrator Interface. Default: IPA.
Options	Appears when Cumulative Statistics are being collected for Socket Management transactions. Note: The Configuration Table Parameters panel shows the option line displayed. If AITSTATOP=YES is not specified in the configuration table, this line does not appear. At the bottom of this panel are the PF key selections for browsing the Socket Management tools defined by this configuration table. PF2 starts the browse of the Listen/Receive tools, and PF6 starts the browse of the Send tools.

Operations

The Operations component of the Administrator Interface provides a collection of functions to diagnose and manage Socket Management. The Socket Management Task-Related User Exit (TRU Exit) provides the call interface to the TCP transport provider address space. This Task-Related User Exit must be enabled to CICS/TS before any transaction can execute a Socket Management call, and when it is disabled to CICS/TS, all TCP and UDP endpoints are terminated.

This Task-Related User Exit is referred to in this section as the *Socket Management Interface*. The process of enabling the TRU Exit is referred to as Socket Management *startup*, and the process of disabling the TRU Exit is referred to as Socket Management *shutdown*.

Operations Control Menu Panel

The Operations Control Menu panel provides functions to diagnose and manage the Socket Management Interface. These functions are summarized first for the menu, and then the panels they invoke are described.

This Operations Control Menu panel is invoked by the O function code from anywhere within the Administrator Interface.

The following is an example of the Operations Control Menu panel.

```
04/27/01 OPERATIONS CONTROL MENU                                OC01
19:10:05  CPT INTERFACE STARTED AT 14:04 ON 04/27/01 IN APPLID CICSQA01

                                PF6) WE: WATCH ERROR LOG

                                PF7) QI: QUIESCE IDLE LISTENERS
                                PF8) QD: DRAIN CPT INTERFACE
                                PF9) SD: SHUTDOWN ALL IMMEDIATE

                                PF10) ZS: RESET ALL STATISTICS
                                PF11) RT: RESET ONLINE TRACE QUEUE
                                PF12) RE: RESET ERROR LOG QUEUE

>
Pf1=?   Pf3=goback   Pf4=Utilization   Pf5=Configuration
```

Operations Functions

The following operation functions can be selected using the PF key shown before it on each line of the menu, or they can be selected by entering the two-character function code shown after the PF key. These function codes can also be entered on other Administrator Interface panels to jump to these operation functions directly without going through this menu panel.

Watch Error Log

Lets the administrator browse the Socket Management error messages that are normally sent to the ACER Transient Data queue. These messages are displayed by a wrap-around queue of Temp Storage records similar to the Online Trace queue.

Quiesce Idle Listeners

Lets the administrator close the endpoints of all currently idle Listener tasks. An idle Listener task is one that currently has no active connections established to its local port number.

Drain Socket Management Interface	<p>Lets the administrator close the endpoints of all Listener tasks to prevent a new connection being accepted from a remote host, and it prevents any CICS/TS task from making a connect call to a remote host.</p> <p>Draining the Socket Management Interface implies that all existing connections are allowed to complete, but establishing a new connection is blocked, in preparation for Socket Management shutdown.</p>
Shutdown All Immediate	<p>Lets the administrator force an abortive close on all active connections, and invoke the Socket Management termination routine, just the same as the IPPR transaction.</p> <p>This action shuts down the Socket Management interface immediately.</p>
Reset All Statistics	<p>Lets the administrator reset both the cumulative statistics and the endpoint statistics that are shown on the Browse Detail and Utilization Summary panels.</p>

Socket Management Interface Error Log Panel

The Socket Management Interface Error Log panel displays the Socket Management error messages normally sent to the ACER Transient Data queue, which is the Socket Management Error Log. These error messages are copied into a temporary storage queue where they can be browsed online. This enables an administrator to diagnose Socket Management error conditions more rapidly by viewing online the messages produced by the Socket Management Interface.

The following is an example of the Socket Management Interface Error Log panel.

```

1  mm/ dd/ yyyy                INTERFACE ERROR LOG 2                3 WE01
   17:41:29

4  00028 T09124I T09CINIT ESTABLISHED SOCKETS COMPATABILITY WITH JOBNAME: NAME002T
5  ***3) T09123I T09CINIT INITIAL WRITE TO ERROR LOG TD QUEUE
   ***4) 6T09111I T09CINIT DEFAULT TRANSLATION TABLE T09XENG LOADED
   ***5) T09147I T09CINIT CA-NetMaster Network management exit successfully enable
00036 T09493I T09TCMDS NetMaster command server started on port 3024
   ***7) T09486I T09TCMDS 04/01/02 10.50 C6006779
   ***8) T09491I T09TCMDS User NAME002 signed on from 141.202.198.171:01126

7  >
9  Pf1=?                        8
   Pf2=Show First   Pf6=Window Left/right   Pf9=Show Last   Pf12=Watch Release

```

Where:

- 1 Current date and time on this CICS/TS system.
- 2 Title of this panel.
- 3 Panel ID and page number.
- 4 CICS/TS task number of the endpoint task generating the traces messages.
- 5 A line count of subsequent messages produced by the same task.
- 6 Socket Management message number for each new trace message.
- 8 The command line used for specifying a function code.
- 10 The message area for command feedback and error notification.
- 11 The PF keys currently active for this panel.

Error Log Queue Description

The Error Log queue is a Temp Storage queue whose size is determined by the AISELOG= parameter of the T09MCICS macro specified in the configuration table. The default size of the Error Log queue is 30 records, and its maximum size is 999 records. When the Error Log queue becomes full, the error messages wrap around and start overlaying the first messages in the queue.

This queue can be changed in size or cleared by the RE (Reset Error) function on the Operations Control Menu panel. If the Error Log queue is set to zero, the following error message appears on the Socket Management Interface Error Log panel:

```
...NO ERROR RECORDS LOGGED...
```

Note: The Error Log queue contains only records from the ACER Transient Data destination. Statistics records sent to ACST and Trace records sent to ACTR destinations are not displayed here, even though these records may be mixed together in a CICS/TS sysout data set.

Browsing the Error Log Queue

PF key selections let the user browse forward and backward through the messages, scroll the display left and right, and jump to the first or last message in the queue. When finished with browsing the error messages, you can use the PF12 key to exit the Socket Management Interface Error Log panel. The PF12 key releases the Error Log queue, and goes to the Utilization Summary panel.

When more than one user is browsing the Error Log queue, only the first user has the PF12 option to release the Error Log queue. The other users will have to use the O function code to return to the Operations Control menu.

Note: New error messages produced while the Socket Management Interface Error Log panel is displayed cannot be added to the Error Log queue. In order to see if any new error messages have arrived, you must use PF12 to release the Error Log queue and exit the panel. Then the WE function can display the Socket Management Interface Error Log panel again.

If the CICS/TS system is brought down and then warm-started, the Temp Storage queue containing the Error Log messages may be kept. The administrator should be aware that error messages displayed on the Socket Management Interface Error Log panel could be from a previous execution of the CICS/TS system, if the Error Log queue was not reset with the RE function.

Quiesce Idle Listeners Function

Entering the QI function code, or using PF7 on the Operations Control Menu panel causes the Quiesce/Close Confirmation panel to appear, prompting for a confirmation of the request with this message:

```
PLEASE USE PF9 (QV) TO CONFIRM REQUEST TO QUIESCE IDLE TASKS!
```

By using the PF9 key or entering the QV function code, the administrator confirms that all idle Listeners tasks will have their endpoints closed and tasks terminated. An idle Listener task is one that currently has no active connections established to its local port number.

If the quiesce request is to be aborted by not confirming on the Quiesce/Close Confirmation panel, you can use PF3 to return to the Operations Control Menu, or you can use any of the standard function codes or PF keys to go to another panel.

Drain Socket Management Interface Function

Entering the QD function code, or using PF8 on the Operations Control Menu causes the Quiesce/Close Confirmation panel to appear, prompting for a confirmation of the request with the following message:

```
PLEASE USE PF9 (QV) TO CONFIRM REQUEST TO DRAIN THE INTERFACE!
```

By using PF9 key or entering the QV function code, the administrator confirms that no new connections can be started by local CICS/TS tasks or from clients on remote hosts. This function causes all Listener tasks to have their endpoints closed and tasks terminated, in preparation for terminating the Socket Management Interface.

If the drain request is aborted by not confirming on the Quiesce/Close Confirmation panel, you can use PF3 to return to the Operations Control Menu, or you can use any of the standard function codes or PF keys to go to another panel.

Shutdown All Immediate Function

Entering the SD function code, or using PF9 on the Operations Control Menu panel causes a single prompt to display on the Operations Control Menu, prompting for a verification of the shutdown with the following message:

```
PLEASE USE PF9 (SV) TO CONFIRM COMPLETE SHUTDOWN REQUEST!
```

By using the PF9 key or entering the SV function code, the administrator confirms that the Socket Management Interface should be terminated and that any remaining connection endpoints will have an abortive close. After entering the confirmation, the next message displayed should report the termination of the Socket Management Interface:

```
T09120I T09TTERM TERMINATION SUCCESSFUL
```

If the shutdown request is aborted by not confirming on the Operations Control Menu, you can use PF3 to return to the Main Menu panel, or you can use any of the standard function codes or PF keys to go to another panel.

Reset All Statistics Function

Entering the ZS function code, or using PF10 on the Operations Control Menu causes the Reset Statistics panel to appear, prompting for a confirmation of the request with this message:

```
PLEASE USE PF9 (ZV) TO CONFIRM REQUEST TO RESET STATISTICS
```

By using the PF9 key or entering the ZV function code, the administrator confirms that both the Cumulative Statistics and the Endpoint Statistics being collected for all endpoints will be cleared.

If the reset request is aborted by not confirming on the Reset Statistics panel, you can use PF3 to return to the Operations Control Menu panel, or you can use any of the standard function codes or PF keys to go to another panel.

Startup Socket Management Interface Function

The Startup function for Socket Management is not normally displayed on the Operations Control Menu, because the Socket Management Interface is usually active when using it. However, if the Socket Management Interface was terminated or did not initialize successfully, this message displays on the Main Menu for the Administrator Interface:

```
T09122E T09AUTIL INTERFACE NOT AVAILABLE (CICS 'AEY9')
```

If the O function code or PF6 is used on the Main Menu at that time, a special version of the Operations Control Menu appears to start up the Socket Management Interface, as shown below.

```
04 / 27 / 01                OPERATIONS CONTROL MENU                OC01
19 : 22 : 08

PLEASE USE PF9 (SU) TO STARTUP CPT

>
Pf1=?   Pf3=goback   Pf4=Utilization   Pf5=Configuration
```

This version of the Operations Control Menu has only one function to confirm the startup of the Socket Management Interface. By using the PF9 key or entering the SU function code, the administrator confirms that the Socket Management Interface should be initialized. This action is equivalent to using the IPST transaction to start the Socket Management Interface.

The next panel to appear is the Utilization Summary panel. The administrator can just use the Enter key on the Utilization Summary panel to see if the Listener tasks defined in the configuration table have started. Then you can use the WE function to go to the Socket Management Interface Error Log panel to check the Socket Management initialization messages that are written to the Socket Management Error Log.

Index

A

active task (AMXT) monitoring, 5-2

Administrator Interface

- accessing through the main menu, C-2
- Browse Detail panel, C-10
- bypassing the Main menu, C-4
- configuration, C-17
- Configuration Table Parameters panel, C-18
- CPT Interface Error Log panel, C-23
- defined, C-1
- definitions of PF keys and navigation codes, C-3
- features, C-1
- leaving, C-4
- Operations component, C-21
- Operations Control menu, C-27
- Operations Control Menu panel, C-22
- Quiesce/Close panel, C-25
- Quiesce/Confirm panel, C-16
- utilization, C-5
- Utilization Summary panel, C-5

automatic Socket Management termination, 2-6

B

batch job, TCPEEP, 7-2, 7-13

Browse Detail panel

- Administrator Interface usage, C-10
- confirming the quiesce, C-16
- displaying
 - active connection endpoints, C-11
 - client UDP transaction detail, C-15
 - endpoint statistics, C-10
 - listener endpoints, C-14
 - server transaction details, C-12
 - UDP endpoints, C-15

- sending an ICMP echo request, C-17
- terminating the endpoint task, C-16

C

CICS

- DCT definition statements for transient data queue names, 5-7
- PCT program control statements, 5-10
- PLT entries for initialization, 5-3
- PPT processing program statements, 5-8
- restricting Socket Management startup and shutdown transaction security, 2-7

Client

- UDP transaction detail-displaying with the Browse Detail panel, C-15
- Utilization Summary details, C-10

Client/Data Listener. *See* Listener

Component Trace, 7-14

configuration table

- Administrator Interface, C-17
- field descriptions, C-19
- Parameters panel, Administrator Interface usage, C-18
- suffixes, 3-2

CPT

- definition statements
 - T09MEND, 4-20
 - T09MLSTN, 4-8, 4-13
- Error Log panel, browsing the Error Log queue, C-24
- Interface Error Log panel
 - Administrator Interface usage, C-23
 - displaying Socket Management error messages, C-23
- macros, instructions, 4-1

CPT API

invoking Listener with translation from Listen Tool, 4-12

D

DCT entries, copying from CPT SAMP library, 5-4, 5-7, 5-8, 5-10, 5-12

definition statements, T09MTRAN, 4-15

E

endpoints

clearing cumulative statistic with the Operations Control Menu panel, C-26

confirming closed

for Listener tasks, C-25

with the Quiesce/Close Confirmation panel, C-25

displaying

data fields for in the Utilization Summary, C-8

details for using the Utilization Summary, C-9

inactive in the Utilization Summary, C-9

listener tasks, C-14

statistics with the Browse Detail panel, C-10

UDP with the Browse Detail panel, C-15

terminating the CICS task with the Browse Detail panel, C-16

viewing active connections with the Browse Detail panel, C-11

Error Log queue

browsing in the CPT Error Log panel, C-24

determining size of, C-24

eTrust

CA-ACF2 configuration, 6-2

CA-Top Secret configuration, 6-3

examples

T09TCMDS

eTRUST CA-ACF2, 6-2

eTRUST CA-Top Secret, 6-3

RACF, 6-4

I

initialization

failure

Task-Related User Exit, 2-4

transient data queues not defined, 2-4

messages written in

Socket Management Transient Data queue, 2-3

system console log, 2-3

initializing Socket Management

Interface with the T09TSTRT program, 2-2

installation. *See Getting Started* for details

L

Listen tool, invoking Listener with translation from, 4-12

Listener

configuring translation in, 4-8

invoking with translation from

Listen tool, 4-12

load module T09CONFIG, 4-1

M

macros

T09END, 4-1

T09MCICS, 4-1

T09MLSTN, 4-1

to configure CPT, 4-1

MXT concurrent tasks

monitoring, 5-2

O

online tracing options using the Utilization Summary, C-10

Operations component Administrator Interface, C-21

Operations Control menu

Administrator Interface usage, C-27

starting the Socket Management Interface, C-27

Operations Control Menu panel
Administrator Interface usage, C-22
confirming Socket Management Interface
termination, C-26
diagnosing and managing the Socket
Management Interface, C-22
functions descriptions, C-22

P

PLTPI table, defining Socket Management
initialization program, 5-2

PLTSD table, defining Socket Management
termination program, 5-2

program statements
CICS PCT, 5-10
CICS PPT, 5-8

Q

Quiesce/Close Confirmation panel
Administrator Interface usage, C-25
confirming
closed endpoints for Listener tasks, C-25
endpoints are closed, C-25

Quiesce/Confirm panel
aborting termination, C-16
Administrator Interface usage, C-16

R

R,TRACE command reply, 7-16

RACF configuration, 6-4

remote host, verifying activation, C-17

S

SCTYEXIT security program, A-1

security
communications block, A-3
logic flow, A-2
program return, A-2
SCTYEXIT program, A-1

Socket Management, A-1

security configuration
eTrust CA-ACF2, 6-2
eTrust CA-Top Secret, 6-3
RACF, 6-4

server transactions, displaying
detail with the Browse Detail panel, C-12
Utilization Summary details, C-10

SIT entries, 5-2

SMP/E
creating translation tables, 3-3
modifying translation tables, 3-3

Socket Management
configuration file, 1-5
definition statements
adding to CICS definition resource tables, 5-1
T09MLSTN, 4-15
displaying transaction summary statistics, C-7
environment

CICS resource definitions, 1-5
overview, 1-5

error messages
displaying with the CPT Interface Error Log
panel, C-23

initialization program, defining in
PLTPI table, 5-2

Interface
CICS transaction security, 2-7
confirming closed with the Operations
Control Menu panel, C-26
diagnosing and managing, C-22
failure during initialization, 2-4
initialization messages, 2-3
initializing with the T09TSTRT program, 2-2
starting with the Operations Control Menu,
C-27

terminating, 2-5
terminating automatically, 2-6
termination messages, 2-6

listing endpoints in the Utilization Summary, C-7
optional security program, A-1
security communications block, A-3
termination program, defining in PLTSD table, 5-2
tools, listen, 1-3
Transient Data queue initialization messages, 2-3

Socket Management Command Server (T09TCMDS), 1-2

Starting the Unicenter NetMaster Administrator
Interface, B-1

statements, Socket Management definition, 5-1

suffixable configuration tables, 3-2
system console log initialization messages, 2-3

T

T09CONFIG load module, 4-1
T09END macro, 4-1
T09MCICS macro, 4-1
T09MEND definition statement, 4-20
T09MLSTN
 definition statement, 4-8, 4-13, 4-15
 macro, 4-1
T09MTRAN transactions, parameter list passed, 4-19
T09TCMDS
 examples
 eTRUST CA-ACF2, 6-2
 eTRUST CA-Top Secret, 6-3
 RACF, 6-4
 security, 6-1
T09TCMDS, Socket Management command server,
1-2
table suffixes, 3-2
TCPEEP
 batch job, 7-2, 7-13
 FORMAT, 7-5
 GROUPS, IUCV or HPNS, 7-9
terminating Socket Management Interface, 2-5, 2-6
termination messages for Socket Management, 2-6

TRACE
 command, 7-14
 CT, 7-14
 REPLY command, 7-16
translation configuring in Listener, 4-8
translation tables
 creating with SMP/E, 3-3
 customizing with SMP/E, 3-3

U

Unicenter NetMaster command processor, 6-1
Unicenter NetMaster for TCP/IP. *See* Unicenter
NetMaster
Unicenter NetMaster Interface command processor, 1-2
utilization
 Administrator
 Interface, C-5
 Administrator Interface panels, C-5
Utilization Summary
 browsing
 client task details, C-10
 server task details, C-10
 displaying
 endpoint data fields, C-8
 endpoint details, C-9
 listing Socket Management endpoints, C-7
Utilization Summary panel
 Administrator Interface usage, C-5
 displaying endpoints in the Administrator
Interface, C-5



Computer Associates™