Diagnostic Procedures

This chapter describes the problem determination and reporting procedures for Cisco IOS for S/390. The chapter organized into the following sections.

- Problem Determination

  Describes problem diagnostic procedures and how to use the diagnostic commands.

- Problem Reporting

  Describes the documentation needed when reporting problems to customer support.

- Troubleshooting GateD

  Describes how to diagnose problems with GateD.

- Troubleshooting the CDLC driver

  Describes procedures to use with the CDLC driver.

- Troubleshooting IUCV Sockets

  Describes procedures specific to IUCV sockets

Problem Determination

If Cisco IOS for S/390 does not respond properly, diagnostic action is required to restore proper service. This section provides a background in performing diagnostic action and gathering documentation for problem resolution.

Although there can be any number of different kinds of software failures, the majority of them fit into some general categories.

Whenever a problem is suspected, a good first step in problem determination is to check the T01LOG and JES message log (or console log) for any messages that may indicate the nature of the problem. The IFS **LOGGING** command can be used to enable more diagnostic messages, such as Debugging messages. For more information, read LOGGING in the Cisco IOS for S/390 Operation chapter of this manual.

Failures That Produce Dumps

In the event of an abend, a system dump (SVC dump) will be produced. This is usually accompanied by message T00IF050E in the message log. Save the dump, logs, and the complete Cisco IOS for S/390 job output, and forward to Customer Support.

For certain abends, the dump is suppressed, for instance, if an external application passes an invalid buffer address. In this case, an abend message may appear in the log without an accompanying dump. If this happens frequently, the application should be identified and corrected.

Failures Due To Network Problems

If the problem is reproducible, use the real-time network trace facility, **TCPEEP**, to diagnose the problem. For intermittent problems, use the **TRACE CT** command to initiate a non-wrapping trace of the network interface(s) involved. For more information, read TCPEEP and TRACE.

Failure of Cisco IOS for S/390 to Respond

Certain problems may lead to a failure of Cisco IOS for S/390, or a major application, to respond. Try to identify the failing component.

Failure of Major Applications

Determine whether the TCP stack is not responding, or whether the problem involves only an application (such as Telnet or FTP). Try to ping the Cisco IOS for S/390 host from a workstation on the local area network. This will tell whether the network interface is alive and the IP layer is responding. If there is no response to the ping, check the status lights on the network device, and note what they indicate. Using a simple application such as the TSO **PING** command to send an outbound request will tell whether the transport provider is active. The **NETSTAT** and **SYSTAT** commands can be used from TSO or the operator console to query and cancel connections. For more information, read PING and NETSTAT/SYSSTAT.

Loops

Cisco IOS for S/390 has loop detection code in most cases, to cancel a process that is looping, and take a diagnostic dump. If Cisco IOS for S/390 is not responding, and using large amounts of CPU, it may be necessary to cancel the job. First take a dump of the address space using the Cisco IOS for S/390 **SVCDUMP** command, or, if that fails to respond, use the console **DUMP** command.

Latch Contention

Cisco IOS for S/390 uses a latch facility, called ILATCH, to serialize use of resources. If a latch or latches are not freed, other processes can be affected. Cisco IOS for S/390 has automatic hung-latch detection code that will normally issue message T00IF038W in this event. Additionally, the **ILATCH DISPLAY** command can be used to show any latches held over 60 seconds. Use the **SVCDUMP** command to take a diagnostic dump of Cisco IOS for S/390. Use the **ILATCH FREE** command to free any held latches, starting at the latch with the highest number, and working down. For more information, read ILATCH.

Storage shortages

If the amount of virtual storage being used by Cisco IOS for S/390 exceeds the threshold defined on the MAXSTGPCT parameter of the IFSPARM statement in the IJTCFGxx configuration member, it will stop accepting new connections. This may be seen by TOPEN error messages in the T01LOG, accompanied by TPL dumps. Check the TPL return code at offset X'10' in the TPL to see whether it indicates a resource shortage condition. (See the *Cisco IOS for S/390 Unprefixed Messages and Codes* for a description of the TPL return codes). This condition may also be accompanied by message T00IF092W on the console.

Use the **IFS POOL** command and **VSM** command to display virtual storage allocation. Use the IFS **SVCDUMP** command to take a diagnostic dump of the Cisco IOS for S/390 address space. In some cases, it may be possible to use **NETSTAT** to free up storage by cancelling old or inactive sessions. Read NETSTAT/SYSSTAT for more information.

If the problem is chronic and involves a slow leak of virtual storage, use the SMF statement of the IJTCFGxx configuration member to turn on INTERVAL recording and record subtype 80. Be sure your system SMF parameters and exits (such as IEFU85) allow the recording of the Cisco IOS for S/390 SMF record type. When the SMF data is collected, the SMF Report Writer Program can be used to view storage utilization and growth during the recording period. Read SMF Report Writer Program for more information.

Note Use of the product ABEND-Aid is not recommended for Cisco IOS for S/390 customers. Dumps formatted by this product do not include many of the control blocks required for problem analysis.

Problem Reporting

When reporting a Cisco IOS for S/390 problem to customer support, document as much information as you can to fully characterize the state of the system at the time of failure and any detectable failure symptoms. The exact documentation collected for a given problem varies, depending on the kind of failure experienced.

In general, the following items should always be provided as the initial documentation of a failure:

- The release and maintenance level of the product and function.

  Note that optional products such as EPS and CPT have their own release levels independent of Cisco IOS for S/390.
- Always include job output. This includes:

  T01LOG

  SYSPRINT

  DNRLOG

  MAPLOG

  SNMLOG
- The console log, or the job log, if running in batch.
- The conditions leading up to the failure.
- Whether the failure is reproducible or a one-time occurrence.
- Whether this is a new failure or if the product was working until this happened (maintenance, more users, new application).

It is better to collect an abundance of initial documentation than wait to have Technical Response Center technicians request additional information later when the data may not be available (especially in the case of intermittent failures or hardware problems).

ABENDs

In the event of an ABEND, collect the following information:

- The SVC dump of the failing address space.
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including the JES log, T01LOG, and any enabled traces. If the ABEND Is in a Cisco IOS for S/390 application (for example, Server Telnet or FTP), and you are running the application in a address space separate from the gateway, collect the job output from the application address space as well.
- A detailed description of the action immediately preceding the failure (if known) or conditions of the system (such as 4500 Telnet sessions).
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace while recreating the failure. If random, you may be able to capture the trace data using the TRACE command to start a wrap-around trace. Read TCPEEP and TRACE for more information.

Incorrect Output – Telnet

For incorrect output through Telnet, collect the following information:

- A detailed description of the action immediately preceding the failure (if known) or conditions of the system (such as 4500 Telnet sessions).
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including the JES log, T01LOG, and any enabled traces.
- Any client software involved, including the name of the vendor and the release number.
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace and a VTAM buffer trace while recreating the failure.

Incorrect Output – FTP

For incorrect output through FTP, collect the following information:

- A detailed description of the action immediately preceding the failure (if known) or conditions of the system (such as 100 terabyte transfer).
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including the JES log, T01LOG, and any enabled traces.
- A log or screen print of the failing transfer, including any messages.
- Any client software involved, including the name of the vendor and the release number, and the direction of transfer and file format; the contents only if requested.
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace.
- You can also set parameters to collect information when running FTP. For FTP2, specify TESTI. For FTP3, you can specify TRACE. Read the *Cisco IOS for S/390 User's Guide* for more information on these commands.

Incorrect Output – DNR, EPS, etc.

For incorrect output through DNR, EPS, etc., collect the following information:

- A detailed description of the action immediately preceding the failure (if known) or conditions of the system.
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including the JES log, T01LOG, and any enabled traces.
- Any error logs and/or job log from the failing component.
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace and a component-specific trace or debug log. Obtain a VTAM buffer trace only if requested.

Hardware

For hardware problems, collect the following information:

- A detailed description of the action immediately preceding the failure (if known) or conditions of the system, and the MVS console log showing Cisco IOS for S/390 messages.
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including the JES log, T01LOG, and any enabled traces.
- The model and microcode version of the network controller and the configuration in the channel path (length, devices in chain, presence of switches, etc.)
- The front-panel display at the time of the failure; note whether the data displayed is static or changing.
- Note whether the failure is reproducible or random. If reproducible, obtain a TCPEEP trace and a GTF channel trace while recreating the failure; if random, notice whether resetting the controller and restarting Cisco IOS for S/390 resolves it.

Startup and Parameter Errors

For startup problems and parameter errors, collect the following information:

- The job output of the Cisco IOS for S/390 gateway at the time of failure, including the JES log, T01LOG, and any enabled traces.
- A listing of the parameter and command members affected (especially any recent changes).
- Whether the failure is reproducible or random.

TCP API Trace Procedures

Use the following steps to gather detailed documentation for API failures or C-sockets interface problems.

Caution You are advised to gather the information all at one time since separate sessions have time stamps which don't match up between components:

Step 1 Raise the size of the Cisco IOS for S/390 internal IFS trace table. This keeps the IFS internal trace table from wrapping too quickly. In the Cisco IOS for S/390 startup JCL there is a CMND parameter representing a member name in the SYSPROC DD library. You can set the IFS trace table to 1 MB by placing this command in the CMND member:

MODIFY TRACE ON SIZE(256)

To use a half megabyte of above the line storage for the IFS trace table, set SIZE(128) on the **MODIFY** command.

Step 2 If a trace address space is not running for the Cisco IOS for S/390 stack, start one now. See the TRACE section in the TSO Diagnostic Commands chapter for more information on the TRACE command and the trace address space.

Step 3 If Cisco IOS for S/390 is not running, start it now.

Step 4 Begin a trace instance using TCPEEP or the TRACE command. (See TCPEEP and TRACE in *Cisco IOS for S/390 System Management Guide*.) Remember, it is always better to capture as much *relevant* data as possible the first time. In order to maximize the amount of relevant data captured, trace filters should be carefully chosen to avoid filling trace data sets with data not related to the problem session or application.

If the problem session is of short duration and is readily reproducible, it may be sufficient to use TCPEEP, or to write a wraparound trace to a relatively small data set, using the TRACE command. If the problem takes longer to develop, you will probably need to use the TRACE command, and write the output to a file that is sufficiently large to hold the data. Specify trace group TLI. If the problem involves network traffic, specify group NETIF. If possible, use filters (host names, port numbers, protocols) to limit the amount of data traced. Only trace the data if necessary.

Step 5 Start the application and wait for the hang and/or failure to occur.

Step 6 From the MVS console, issue the command:

F *runtcp***,SVCDUMP,JOB(***jobname***)**

where *runtcp* is the name of the Cisco IOS for S/390 stack, and *jobname* is the job name of the API application. This will take a single SVC dump of both address spaces.

Step 7 Stop the trace or TCPEEP job

Step 8 Send everything you've collected to Customer Support:

- All the Cisco IOS for S/390 job output including JES logs
- The SVC dump of the Cisco IOS for S/390 address space
- All the job output from the failing application address space
- The SVC dump of the failing application address space
- The output from the TRACE or TCPEEP command

Troubleshooting GateD

This section provides information on troubleshooting GateD. For more information on GateD commands, refer to the *Cisco IOS for S/390 User's Guide*.

GateD Logs

There are three logs associated with GateD:

GateD Log	Description
GTDLOG	Normal GateD messages are written to this log.
GTDERR	GateD error messages with a level greater than the syslog up to level are written to this log.
GTDTRC	GateD trace messages are written to this log.

The OPSFmon and ripquery tools are provided with Cisco IOS for S/390 to help with GateD implementation. Read the OSPFMON and RIPQUERY sections in the Diagnostic Procedures chapter of this manual for more information.

Controlling GateD

ACTEST can be used to stop GateD. To determine the *pta_address*, issue the command **TASK GATED** and note the *pta_address* that is returned. Then issue the command **PPOST** *pta_address* **CLOSE**. GateD will stop after this command is issued, but will then restart automatically, using the configuration member specified on the IP statement in member TCPCFGxx.

Creating Formatted Dumps for GateD

GateD has a trace dump facility which formats many of the control blocks for use in debugging. You can PPOST the GateD task via ACTEST to signal it to perform special actions. These are:

Command	Description
PPOST gated-pta OUTPUT	Generate formatted dump.
PPOST gated-pta INPUT	Reinitialize GateD (in other words reread config).
PPOST gated-pta ATTN	Suspend or resume tracing (in other words toggle).
PPOST gated-pta USER	Perform immediate rescan of interfaces.
PPOST gated-pta CLOSE	Shutdown GateD (no change).

For more information about RIP and OSPF, refer to RFCs **1058** and **1247**, respectively.

Troubleshooting the CDLC driver

Certain types of CDLC controllers, such as the 3745, are governed by NCP generation parameters and may be subject to VTAM commands dealing with PU activation and deactivation. If the CDLC controller has no NCP dependencies, no VTAM considerations are applicable.

Prior to Cisco IOS for S/390 startup, the channel adapter PU dedicated for IP traffic should be in a PCTD2 state. After Cisco IOS for S/390 initialization, the PU should go into an ACTIV state. When Cisco IOS for S/390 is brought down, the following VTAM message appears and the PU returns to the PCTD2 state:

```
IST259I INOP RECEIVED FOR puname
```

If, upon starting Cisco IOS for S/390, you encounter this message:

```
T01LL106E Device dev_name: initialization sequence failed to complete (timeout).
    Device shutdown initiated.
```

use these commands to reset the channel adapter PU that is dedicated to IP traffic:

```
v net,inact,id=puname
v net,act,id=puname
```

You should then receive the following message:

```
T01LL181I Device dev_name: interface if_name is fully operational.
```

Cisco IOS for S/390 need not be shut down to issue the vary commands.

Check the job output from the T01LOG and the MVS syslog output for CDLC driver ABENDs, device I/O errors, device allocation errors, and other such errors.

Troubleshooting IUCV Sockets

The IUCV sockets replace the corresponding IBM TCP/IP IUCV transport address space and TCP/IP facilities and let existing IUCV-based applications execute transparently. Debugging facilities that exist within those interfaces, as well as IUCV socket traces, can be used to diagnose problems that may appear when using this feature.

Potential problems fall into two broad groups:

- Startup of the IUCV sockets and interface to Cisco IOS for S/390.
- Application-level problems seen during normal operations.

Diagnosing Application-Level Problems

Once the IUCV and Cisco IOS for S/390 tasks have started, application programs can connect to the network via TCP/IP. At this point, problems tend to be protocol related and there are two levels to consider:

- The IUCV transport
- The socket function calls

The socket libraries (IBM C socket library), and the macro socket and IUCV APIs, depend on the correct encapsulation of a socket-family call inside an IUCV parameter list, as well as a functional IUCV connection (path) between the application, the IUCV task, and the Cisco IOS for S/390 task.

IUCV Transport

The application program using IUCV sockets uses two items of information to build a connection with Cisco IOS for S/390 over IUCV (this is different than a socket connection, described in detail below).

- The IUCV subsystem name; by default, VMCF.
- The TCP/IP jobname; by default, TCPIP.

If your configuration matches these values, then connection should proceed without any error. However, the most commonly seen error code is in response to socket function calls. For example:

```
RC=1 on IUCV_CONNECT to tcpip, FD = -254, Path = nn, IPRCODE = 11
```

The IPRCODE of 11 indicates that the task name tcpip was not found or did not accept the connection. Verify that the stepname of the Cisco IOS for S/390 task matches the TCPJOBNAME of the socket library (the Macro API determines this name from a zappable constant in the Cisco IOS for S/390 version of EZASOK03).

It is more difficult to diagnose an incorrect subsystem name because many applications expect that the VMCF subsystem is always present and these applications are not prepared to handle error cases.

These return codes are set by the IUCV sockets support routines when the subsystem is not found or not initialized:

• SNMPGPCN	4 (not found) or 8 (not valid)
• IUCVMULT	10103 (PC numbers not set by SNMPGPCN)
• IUCVSTRT/MAST	Same as that set by IUCVMULT

Note Actions by the IBM C socket library when the IUCV subsystem is not present or is not named VMCF are unpredictable.

Socket Function Calls

Once the IUCV connection has been made and a path exists to Cisco IOS for S/390, the actual socket function can proceed. The number and order of function calls depends on the application, but typically a socket() is allocated. For a server program, this is followed by a bind() and listen(); for a client program, it is followed by connect(). In this example, the function is a socket connect, which takes place over an existing IUCV connection.

Problem determination at this level depends on tracking the function requested by the application, through the socket library, through IUCV, through Cisco IOS for S/390, and out to the network, then following the return status back through the same series of components to the application. The IUCV sockets supply some of the trace tools needed; the remainder exist either in the application, in the socket library, or on the network.

Applications may have some level of socket-call-level tracing that can be enabled via parameters or environment variables.

The IBM C socket library provides the SOCKDEBUG option in hlq.TCPIP.DATA to list the IUCV arguments and reply data to the SYSPRINT DD statement.

The Cisco IOS for S/390 Macro API has a trace function that is enabled by zapping constants in the option module T02U80EZ. It then prints trace records to the designated DD statement. See Macro API Diagnostic Facilities for details of the trace.

Both the IUCV task (RUNIUCV) and the Cisco IOS for S/390 task (RUNTCP) record trace data into an internal trace buffer when the TRACE is turned on and the TEST attribute is set to **ON** for the following task groups:

• IUCV	IJT task group
• TCP	TCP task group

This is accomplished via command statements in the START*xx* script executed at task startup. Read the STARTxx Configuration section in the Cisco IOS for S/390 Operation chapter for detailed syntax of these statements.The samples provided with this feature are pre-configured with usable defaults.

Trace entries are written into the internal circular buffer once tracing has been enabled. They will be formatted to sysout when the following operator command is entered:

- For IUCV, enter:

  f *runiucv*,**ijt snap all**
- For Cisco IOS for S/390, enter:

  f *runtcp*,**tcp snap all**

Syntax Description

runiucv	IUCV address space name
runtcp	Cisco IOS for S/390 address space name

The trace table being snapped exists in different task groups in the IUCV and TCP tasks. In addition to the trace table, selected control blocks will also be formatted.

Trace entries contain details about conditions within the task which include control block addresses, register locations, and internal error indicators. Trace entry format is shown in the "IFS Internal Trace Entries" chapter of "Cisco IOS for S/390 Unprefixed Messages and Codes," however complete interpretation of the trace data is intended to be performed by Customer Support.

Cisco IOS for S/390 can record the network message exchange via the TCPEEP tracing tool. For socket applications, select the TCP level of recording with the DATA option greater than 256.

When possible, collect traces at all points in the application-to-network path from a known, repeatable set of starting conditions. All traces should record the same event, with consistent time stamps. Accurate trace data is key to diagnosing protocol related errors.

Diagnosing Other IUCV Failures

Other than the problems noted above, program checks (ABENDs) can occur or performance issues may be detected during otherwise normal operation.

In the case of ABENDs, it is critical to retain all the memory dump information, along with the sysout logs from *both* Cisco IOS for S/390 and IUCV tasks. If the tasks remain functional, issue an IJT SNAP ALL or TCP SNAP ALL to the IUCV task and Cisco IOS for S/390 task, respectively. In many cases, an SVC dump will be taken automatically of all three address spaces (application, IUCV, Cisco IOS for S/390) and recorded into a SYS1.DUMPxx data set. This dump is intended to be analyzed via the IPCS tool, so formatted printing is not recommended. If possible, note the application(s) running at the time and any other related messages. If an application-task dump occurs, retain that also.

Performance problems are diagnosed by obtaining IUCV and Cisco IOS for S/390 internal traces, which contain very accurate time stamps, along with a TCPEEP trace to monitor the data flow.

Reporting IUCV Failures

When a failure occurs, collect documentation as appropriate (see the previous sections for details) and contact Customer Support. They may request that you send the items of documentation in machine-readable form (in other words, not on paper). More rapid diagnosis can be made if documentation can be analyzed with online tools.

Be prepared to describe the environment that existed at the time of the failure and if the failure is repeatable (and if so, under what circumstances) or if it is an isolated occurrence.

Macro API Diagnostic Facilities

The configuration and diagnostic capabilities available with the Macro API runtime support module, EZASOK03, are described in this section.

The load module T02U80EZ (alias EZASOK03) must be in an APF authorized library. Several application defaults can be set or overridden by zapping the defaults CSECT IUCVCONS in module T02UIUCV.

Output from the Macro API Trace is written to the DD name chosen (default SYSPRINT) in the client job or address space.

This table describes the values of the CSECT IUCVCONS:

Table 4-1 CSECT IUCVCONS

Option Name	Offset	Value	Description
TCPIPJOBNAME	8(x’08’)	CL8’ ’	As delivered, EZASOK03 lets the application set this value with the INITAPI call. If this field is zapped to some other value, the INITAPI value will be overridden at run time.
DNRSSID	16(x’10’)	CL4’ACSS’	This value is only used if the application uses the TYPE=GETHOSTBYADDR or TYPE=GETHOSTBYNAME functions.
VMCFSSID	20(x’14’)	CL4’VMCF’	VMCF subsystem ID
TRACEFLAG	24(x’18’)	X’00’	As delivered, tracing is turned off. X'80' = TRACING ON
TRACESIZE	26(x’1A’)	H’25’	As delivered, the default trace table size is limited to 25 entries before wrapping. This value is only used if the TRACEFLAG value is set to ON.
TRACEDD	28(x’1C’)	C’SYSPRINT’	As delivered, the default trace DD statement name is SYSPRINT. This value is only used if the TRACEFLAG value is set to ON. **Note:** If the OPEN fails for this DD statement, no tracing will be attempted.

Note If you are going to change the Cisco IOS for S/390 jobname etc., use usermod MU1IUCV, found in the SAMP library. It is the preferred way to use SMP/E for zapping.

Examples

Examples of the character formatted socket trace and the in-memory trace table are in the IUCVNOTE member of the SAMP library.

Example

This job is an example of how to zap application defaults/overrides:

```
//*
//* ZAP FOR SETTING EZASOK03 CONFIGURATION OVERRIDES/DEFAULTS
//*
//STEP EXEC   PGM=AMASPZAP
//SYSPRINT        DD   SYSOUT=*
//SYSLIB          DD   DSN=TCPICS.Vxxx.LOAD,DISP=SHR
//SYSIN           DD  *
  NAME  T02UIUCV IUCVCONS
  REP   0008   E3C3D7C9D7404040        'TCPIP  ' - TCPIP ADDRESS SPACE NAME
  REP   0010   C1C3E2C8                'ACSH'    - DNR SUBSYSTEM ID
  REP   0014   C9D5D3D2                'INLK'    - VMCF SUBSYSTEM ID
  REP   0018   80                      TRACE ON
  REP   001A   0032                    50 INCORE TRACE TABLE ENTRIES
  REP   001C   E2E8E2E3D9C1C3C5        SYSTRACE - DD STATEMENT FOR
*                                      CHAR TRACE
/*
```

The tracing output that is made available by zapping T02UIUCV comes in two parts: an in-memory trace table that records the parameter lists, and a character formatted socket trace.

USERMOD

```
++ APAR (MU1IUCV)                                 /* TCP/IP CORRECTIVE MAINTENANCE */
++ VER (Z038)
   FMID(TCP0xxx)
   PRE(TPXXXXX)
 /*

   PROBLEM DESCRIPTION (PSR052177):
   *================================================================================*
   * PROBLEM SUMMARY:                                                               *
   *   ZAP IUCV CONSTANTS FOR ALL MODULES                                           *
   *--------------------------------------------------------------------------------*
   * RECOMMENDATION: APPLY THE APAR TO CHANGE THE CONSTANTS.                        *
   *--------------------------------------------------------------------------------*
   * PROBLEM DESCRIPTION: CHANGE IUCV CONSTANTS                                     *
   *================================================================================*
   *   CJB *

   THE FOLLOWING MODULES/MACROS ARE AFFECTED BY THIS PTF:

     LIST BEGIN
        T02UIUCV
     LIST END

 */ .
++ZAP(T02UIUCV) DISTLIB(ATCPLOAD) .
  NAME T02UIUCV IUCVCONS
  REP 0008 E3C3D7E5F4F1F040                 TCPV410
  REP 0010 C1C3E2E9                         ACSZ
  REP 0014 C9E4C3E5                         IUCV
  REP 0018 8000                             X'80' TRACE FLAG ON
  REP   001A    0032                    50 INCORE TRACE TABLE ENTRIES
  REP   001C    E2E8E2E3D9C1C3C5        SYSTRACE - DD STATEMENT
//SMPCNTL  DD  *
 SET BDY(GLOBAL) .
 RECEIVE S(MU1IUCV) .
 SET BDY(TCPTZN) .
 APPLY   S(MU1IUCV) .
//
```

Character Formatted Socket Trace

There are two trace records written to the trace DD statement for each function being traced. The first record, Table 4-2, is written before the PC call, the call to IUCVMULT, or the dirsrv request (get...by... emulation). The second record, Table 4-3, is written after the function has completed.

Table 4-2 Character Formatted Socket Trace

Field #	Description
1	The subtask name, which is supplied by the Macro API caller on the INITAPI call.
2	The socket function, IUCV function, or dirsrv request.
3	The IUCV path (if applicable).
4	The value in IPTRGCLS+2, which is usually the socket number (if applicable).
5	The return code. On successful calls this field often contains socket call result information. For example, on a successful read(), this field contains the number of bytes read. On unsuccessful socket calls, this field contains -1.

Table 4-2 Character Formatted Socket Trace (Continued)

Field #	Description
6	If Field #5 contains -1, Field #6 will contain the errno value (whose meaning can be found in the header library member TCPERRNO). On a successful INITIAL MESSAGE call, Field #6 contains the maximum socket value returned by TCP.
7	Field #7 in the character formatted socket trace corresponds to the trace entry type that gets generated and written to the in-memory table. The TYP5 record on an IUCVMINI SET function gives the address of the corresponding in-memory trace table. There is a separate in-memory trace table for every TCB (TASK=). The address on the TYP5 record can be used to find the correct in-memory table when a dump of the application region is taken.

Table 4-3 Format of the In-Memory Trace Table Records

Field #	Description
1	7 byte TRACTYP + 1 byte to indicate entry type with: 0=IUCVSEND (usually a socket function) before PC call 1=IUCVSEND after PC call completes 2=DIRSRV request before execution 3=DIRSRV request after execution 4=IUCVMINI SET before call to IUCVMULT 5=IUCVMINI SET after call to IUCVMULT 6=IUCVMINI CLR before call to IUCVMULT 7=IUCVMINI CLR after call to IUCVMULT 8=IUCVCONN CALL before call to IUCVMULT 9=IUCVCONN CALL after call to IUCVMULT
2	Current IPARML (40 bytes) or DPL (56 bytes).
3	External interrupt ECB or continuation of DPL.
4	First 12 bytes of the external interrupt buffer or continuation of DPL.
5	UPRM literal.
6	Address of calling program’s parameter list.
7	Double word STCK value.

The in-memory trace table has a default size of 25 entries and will wrap.

Examples of the character formatted socket trace and the in-memory trace table are in the IUCVNOTE member of the SAMP library.

IUCV Debugging Checklist

Use this checklist to help you gather information necessary for reporting failures before you call Customer Support.

- Make sure you are using the Cisco IOS for S/390 version of IUCVMULT. This is described in IUCV Socket Compatibility in the IUCV Socket chapter.
- IUCV address space with IJT SNAP dump (**f** *runiucv*,**ijt snap all**).
- Cisco IOS for S/390 address space with TCP SNAP dump (**f** *runtcp*,**tcp snap all**).

- Application address space, with trace if possible.
- SVC dump, if available.
- Memory allocation dump from both IUCV and Cisco IOS for S/390 address spaces if you suspect a storage shortage (**f** *runtcp*,**d pool**).