z/OS Communications Server

# SNA Diagnosis Volume 2: FFST Dumps and the VIT

*Version 1  Release 9*

z/OS Communications Server

# SNA Diagnosis Volume 2: FFST Dumps and the VIT

*Version 1  Release 9*

# Contents

# Figures

**xiii**

—

# Tables

# About this document

This document is intended to help system programmers in a VTAM® environment diagnose problems with the VTAM program. Use the document to isolate and identify problems with your VTAM network and to collect appropriate documentation to resolve network problems. This document supports both IPv6 and IPv4. Unless explicitly noted, information describes IPv4 networking protocol. IPv6 support is qualified within the text.

## Who should read this document

System programmers should use this document to analyze a VTAM problem, classify the problem as a specific type, and provide information about the problem to an IBM® Support Center representative.

You should be familiar with the VTAM service aids and the procedures for reporting problems to an IBM Support Center representative.

## How this document is organized

This document is organized into the following topics:
- Chapter 1, "Using FFST dumps," on page 1 describes dump procedures.
- Chapter 2, "Using the VTAM internal trace (VIT)," on page 9 describes how to use the VTAM Internal Trace (VIT).
- Appendix A, "VTAM internal trace (VIT) record descriptions," on page 27 describes the VIT records.
- Appendix C, "First Failure Support Technology (FFST) probes," on page 429 describes the FFST™ probes that trigger dumps when an unusual condition occurs in VTAM.
- Appendix D, "Communications storage manager (CSM) FFST probes," on page 471 describes the CSM FFST probes that trigger dumps when an unusual condition occurs in CSM.
- "Architectural specifications" lists documents that provide architectural specifications for the SNA protocol.
- "Information APARS and technotes" lists information apars for SNA-related books.
- "Accessibility" describes accessibility features to help users with physical disabilities.
- "Notices" on page 481 contains the notices for this licensed document.
- "Bibliography" contains descriptions of the documents in the z/OS® Communications Server library.

### How to use this document

Use this document to isolate and identify problems with your VTAM network and to collect appropriate documentation to resolve network problems.

#### Determining whether a publication is current

As needed, IBM updates its publications with new and changed information. For a given publication, updates to the hardcopy and associated BookManager® softcopy

are usually available at the same time. Sometimes, however, the updates to hardcopy and softcopy are available at different times. The following information describes how to determine if you are looking at the most current copy of a publication:

- At the end of a publication's order number there is a dash followed by two digits, often referred to as the dash level. A publication with a higher dash level is more current than one with a lower dash level. For example, in the publication order number GC28-1747-07, the dash level 07 means that the publication is more current than previous levels, such as 05 or 04.

- If a hardcopy publication and a softcopy publication have the same dash level, it is possible that the softcopy publication is more current than the hardcopy publication. Check the dates shown in the Summary of Changes. The softcopy publication might have a more recently dated Summary of Changes than the hardcopy publication.

- To compare softcopy publications, you can check the last two characters of the publication's file name (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

### How to contact IBM service

For immediate assistance, visit this Web site:

http://www.software.ibm.com/network/commserver/support/

Most problems can be resolved at this Web site, where you can submit questions and problem reports electronically, as well as access a variety of diagnosis information.

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-IBM-SERV). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

If you would like to provide feedback on this publication, see "Communicating Your Comments to IBM" on page 505.

## Conventions and terminology used in this document

Commands in this book that can be used in both TSO and z/OS UNIX® environments use the following conventions:

- When describing how to use the command in a TSO environment, the command is presented in uppercase (for example, NETSTAT).
- When describing how to use the command in a z/OS UNIX environment, the command is presented in bold lowercase (for example, **netstat**).
- When referring to the command in a general way in text, the command is presented with an initial capital letter (for example, Netstat).

All of the exit routines described in this document are *installation-wide exit routines*. You will see the installation-wide exit routines also called installation-wide exits, exit routines, and exits throughout this document.

The TPF logon manager, although shipped with VTAM, is an application program. Therefore, the logon manager is documented separately from VTAM.

Samples used in this book might not be updated for each release. Evaluate a sample carefully before applying it to your system.

For definitions of the terms and abbreviations used in this document, you can view the latest IBM terminology at the IBM Terminology Web site.

## Clarification of notes

Information traditionally qualified as **Notes** is further qualified as follows:

**Note** Supplemental detail

**Tip** Offers shortcuts or alternative ways of performing an action; a hint

**Guideline**
Customary way to perform a procedure

**Rule** Something you must do; limitations on your actions

**Restriction**
Indicates certain conditions are not supported; limitations on a product or facility

**Requirement**
Dependencies, prerequisites

**Result** Indicates the outcome

## Prerequisite and related information

z/OS Communications Server function is described in the z/OS Communications Server library. Descriptions of those documents are listed in "z/OS Communications Server information" on page 491, in the back of this document.

## Required information

Before using this product, you should be familiar with TCP/IP, VTAM, MVS™, and UNIX System Services.

## Related information

This section contains subsections on:
- "Softcopy information" on page xx
- "Other documents" on page xx
- "Redbooks" on page xxi
- "Where to find related information on the Internet" on page xxi
- "Using LookAt to look up message explanations" on page xxiii
- "Using IBM Health Checker for z/OS" on page xxiv

## Softcopy information

Softcopy publications are available in the following collections:

| Titles | Order Number | Description |
|---|---|---|
| *z/OS V1R9 Collection* | SK3T-4269 | This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R9, in both BookManager and PDF formats. |
| *z/OS Software Products Collection* | SK3T-4270 | This CD includes, in both BookManager and PDF formats, the libraries of z/OS software products that run on z/OS but are not elements and features, as well as the *Getting Started with Parallel Sysplex*® bookshelf. |
| *z/OS V1R9 and Software Products DVD Collection* | SK3T-4271 | This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager and PDF format. This collection combines SK3T-4269 and SK3T-4270. |
| *z/OS Licensed Product Library* | SK3T-4307 | This CD includes the licensed documents in both BookManager and PDF format. |
| *IBM System z Redbooks Collection* | SK3T-7876 | The Redbooks selected for this CD series are taken from the IBM Redbooks inventory of over 800 books. All the Redbooks that are of interest to the zSeries platform professional are identified by their authors and are included in this collection. The zSeries subject areas range from e-business application development and enablement to hardware, networking, Linux, solutions, security, parallel sysplex, and many others. |

## Other documents

For information about z/OS products, refer to *z/OS Information Roadmap* (SA22-7500). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, as well as describing each z/OS publication.

Relevant RFCs are listed in an appendix of the IP documents. Architectural specifications for the SNA protocol are listed in an appendix of the SNA documents.

The following table lists documents that might be helpful to readers.

| Title | Number |
|---|---|
| *DNS and BIND*, Fourth Edition, O'Reilly and Associates, 2001 | ISBN 0-596-00158-4 |
| *Routing in the Internet*, Christian Huitema (Prentice Hall PTR, 1995) | ISBN 0-13-132192-7 |
| *sendmail*, Bryan Costales and Eric Allman, O'Reilly and Associates, 2002 | ISBN 1-56592-839-3 |
| *SNA Formats* | GA27-3136 |
| *TCP/IP Illustrated, Volume I: The Protocols*, W. Richard Stevens, Addison-Wesley Publishing, 1994 | ISBN 0-201-63346-9 |
| *TCP/IP Illustrated, Volume II: The Implementation*, Gary R. Wright and W. Richard Stevens, Addison-Wesley Publishing, 1995 | ISBN 0-201-63354-X |
| *TCP/IP Illustrated, Volume III*, W. Richard Stevens, Addison-Wesley Publishing, 1995 | ISBN 0-201-63495-3 |
| *TCP/IP Tutorial and Technical Overview* | GG24-3376 |

| Title | Number |
|---|---|
| *Understanding LDAP* | *SG24-4986* |
| *z/OS Cryptographic Service System Secure Sockets Layer Programming* | *SC24-5901* |
| *z/OS Integrated Security Services LDAP Client Programming* | *SC24-5924* |
| *z/OS Integrated Security Services LDAP Server Administration and Use* | *SC24-5923* |
| *z/OS JES2 Initialization and Tuning Guide* | *SA22-7532* |
| *z/OS Problem Management* | *G325-2564* |
| *z/OS MVS Diagnosis: Reference* | *GA22-7588* |
| *z/OS MVS Diagnosis: Tools and Service Aids* | *GA22-7589* |
| *z/OS MVS Using the Subsystem Interface* | *SA22-7642* |
| *z/OS Program Directory* | *GI10-0670* |
| *z/OS UNIX System Services Command Reference* | *SA22-7802* |
| *z/OS UNIX System Services Planning* | *GA22-7800* |
| *z/OS UNIX System Services Programming: Assembler Callable Services Reference* | *SA22-7803* |
| *z/OS UNIX System Services User's Guide* | *SA22-7801* |
| *z/OS XL C/C++ Run-Time Library Reference* | *SA22-7821* |
| *System z9 and zSeries OSA-Express Customer's Guide and Reference* | *SA22-7935* |

## Redbooks

The following Redbooks™ might help you as you implement z/OS Communications Server.

| Title | Number |
|---|---|
| *Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 1: Base Functions, Connectivity, and Routing* | *SG24-7339* |
| *Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 2: Standard Applications* | *SG24-7340* |
| *Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 3: High Availability, Scalability, and Performance* | *SG24-7341* |
| *Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 4: Policy-Based Network Security* | *SG24-7342* |
| *IBM Communication Controller Migration Guide* | *SG24-6298* |
| *IP Network Design Guide* | *SG24-2580* |
| *Managing OS/390® TCP/IP with SNMP* | *SG24-5866* |
| *Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender* | *SG24-5957* |
| *SecureWay Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements* | *SG24–5631* |
| *SNA and TCP/IP Integration* | *SG24-5291* |
| *TCP/IP in a Sysplex* | *SG24-5235* |
| *TCP/IP Tutorial and Technical Overview* | *GG24-3376* |
| *Threadsafe Considerations for CICS* | *SG24-6351* |

## Where to find related information on the Internet

**z/OS**

This site provides information about z/OS Communications Server release availability, migration information, downloads, and links to information about z/OS technology

http://www.ibm.com/servers/eserver/zseries/zos/

**z/OS Internet Library**

Use this site to view and download z/OS Communications Server documentation

http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

**IBM Communications Server product**

The primary home page for information about z/OS Communications Server

http://www.software.ibm.com/network/commserver/

**IBM Communications Server product support**

Use this site to submit and track problems and search the z/OS Communications Server knowledge base for Technotes, FAQs, white papers, and other z/OS Communications Server information

http://www.software.ibm.com/network/commserver/support/

**IBM Systems Center publications**

Use this site to view and order Redbooks, Redpapers, and Technotes

http://www.redbooks.ibm.com/

**IBM Systems Center flashes**

Search the Technical Sales Library for Techdocs (including Flashes, presentations, Technotes, FAQs, white papers, Customer Support Plans, and Skills Transfer information)

http://www.ibm.com/support/techdocs/atsmastr.nsf

**RFCs**

Search for and view Request for Comments documents in this section of the Internet Engineering Task Force Web site, with links to the RFC repository and the IETF Working Groups Web page

http://www.ietf.org/rfc.html

**Internet drafts**

View Internet-Drafts, which are working documents of the Internet Engineering Task Force (IETF) and other groups, in this section of the Internet Engineering Task Force Web site

http://www.ietf.org/ID.html

Information about Web addresses can also be found in information APAR II11334.

**Note:** Any pointers in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

## DNS Web sites

For more information about DNS, see the following USENET news groups and mailing addresses:

**USENET news groups**
    comp.protocols.dns.bind

**BIND mailing lists**
    http://www.isc.org/ml-archives/

   **BIND Users**
      • Subscribe by sending mail to bind-users-request@isc.org.
      • Submit questions or answers to this forum by sending mail to
        bind-users@isc.org.

   **BIND 9 Users (This list might not be maintained indefinitely.)**
      • Subscribe by sending mail to bind9-users-request@isc.org.
      • Submit questions or answers to this forum by sending mail to
        bind9-users@isc.org.

## Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM
messages you encounter, as well as for some system abends and codes. Using
LookAt to find information is faster than a conventional search because in most
cases LookAt goes directly to the message explanation.

You can use LookAt from these locations to find IBM message explanations for
z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and
Linux™:

• The Internet. You can access IBM message explanations directly from the LookAt
  Web site at www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.

• Your z/OS TSO/E host system. You can install code on your z/OS systems to
  access IBM message explanations using LookAt from a TSO/E command line
  (for example: TSO/E prompt, ISPF, or z/OS UNIX System Services).

• Your Microsoft® Windows® workstation. You can install LookAt directly from the
  z/OS Collection (SK3T-4269) or the *z/OS and Software Products DVD Collection*
  (SK3T-4271) and use it from the resulting Windows graphical user interface
  (GUI). The command prompt (also known as the DOS > command line) version
  can still be used from the directory in which you install the Windows version of
  LookAt.

• Your wireless handheld device. You can use the LookAt Mobile Edition from
  www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html with a
  handheld device that has wireless access and an Internet browser (for example:
  Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for
  Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows
workstation from:

• A CD-ROM in the z/OS Collection (SK3T-4269).

• The *z/OS and Software Products DVD Collection* (SK3T-4271).

• The LookAt Web site (click **Download** and then select the platform, release,
  collection, and location that suit your needs). More information is available in
  the LOOKAT.ME files available during the download process.

## Using IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework. This book might refer to checks or messages associated with this component.

For additional information about checks and about IBM Health Checker for z/OS, see *IBM Health Checker for z/OS: User's Guide*. Starting with z/OS V1R4, z/OS users can obtain the IBM Health Checker for z/OS from the z/OS Downloads page at http://www.ibm.com/servers/eservers/zseries/zos/downloads/.

SDSF also provides functions to simplify the management of checks. See *z/OS SDSF Operation and Customization* for additional information.

## How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document or any other z/OS Communications Server documentation:

- Go to the z/OS contact page at:

  http://www.ibm.com/servers/eserver/zseries/zos/webqs.html

  There you will find the feedback page where you can enter and submit your comments.

- Send your comments by e-mail to comsvrcf@us.ibm.com. Be sure to include the name of the document, the part number of the document, the version of z/OS Communications Server, and, if applicable, the specific location of the text you are commenting on (for example, a section number, a page number or a table number).

# Summary of changes

This document contains information previously presented in GC31-6851-01, which supports z/OS Version 1 Release 8.

The information in this document includes descriptions of support for both IPv4 and IPv6 networking protocols. Unless explicitly noted, descriptions of IP protocol support concern IPv4. IPv6 support is qualified within the text.

**New information**
- Add group names to simplify turning on VTAM internal trace, see Table 4 on page 17.
- HPR enhancements, see "HCL2 entry for HPR clock event (Part 2)" on page 210.
- "CDN2 entry for CIDCTL NEXTNODE FIND (Part 2)" on page 106

**Changed information**
- Add group names to simplify turning on VTAM internal trace, see:
  - "Selecting trace options" on page 10
  - "Deactivating the VTAM internal trace" on page 25
- HPR enhancements, see:
  - "FBLK entry for FREEBLK macro (Part 1)" on page 191
  - "GBLK entry for GETBLK macro (Part 1)" on page 198
  - "HPR2 entry for HPRCTL macroinstruction (Part 2)" on page 216
  - "RTP entry for RTP PAB dispatch" on page 320
  - "RTPP entry for path switch processing" on page 325
- "RSCx entry for RSCVSCAN macroinstruction (Part 1)" on page 318
- "RTP2 entry for RTP PAB dispatch (Part 2)" on page 322

**Deleted information**
- The APPC Application Suite is removed from the z/OS V1R9 Communications Server product and therefore documentation describing APPC Application Suite support has been deleted.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You might notice changes in the style and structure of some content in this document–for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

**Summary of changes**
**for GC31-6851-01**
**z/OS Version 1 Release 8**

This document contains information previously presented in GC31-6851-00, which supports z/OS Version 1 Release 7.

The information in this document includes descriptions of support for both IPv4 and IPv6 networking protocols. Unless explicitly noted, descriptions of IP protocol support concern IPv4. IPv6 support is qualified within the text.

**New information**
- SNA trace enhancements
    - "COP2 entry for internal trace statistics (Part 2)" on page 153
    - "GBL3 entry for GETBLK macro (Part 2)" on page 201
- Allow wildcards on Enterprise Extender network management interface filters, see "HPR4 entry for HPRCTL macroinstruction (Part 4)" on page 218.
- Interchange node selection function for DSME, see "Directory services probes" on page 442.

**Changed information**
- SNA trace enhancements
    - "COPY entry for internal trace statistics (Part 1)" on page 153
    - "FBLK entry for FREEBLK macro (Part 1)" on page 191
    - "GBLK entry for GETBLK macro (Part 1)" on page 198
    - "GBL2 entry for GETBLK macro (Part 2)" on page 200
    - "XCS2 entry for send message (Part 2)" on page 403
- Sysplex subplexing
    - "GNAM entry for GNAME macro invoked (Part 1)" on page 204
    - "SPT entry for SPT macro invoked (Part 1)" on page 342
    - "XCFJ entry for join XCF group (Part 1)" on page 395
    - "XCFM entry for XCF group member (Part 1)" on page 399
- OSA-Express network traffic analyzer, see "ODP2 entry for OSA-Express QDIO or HiperSockets packets (Part 2)" on page 279.
- APPN route selection trace, see "The TRS common topology trace table" on page 422.

**Deleted information**
- AnyNet® function is removed from the z/OS Communications Server product and therefore documentation describing AnyNet support has been deleted.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You might notice changes in the style and structure of some content in this document–for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

**Summary of changes**
**for GC31-6851-00**
**z/OS Version 1 Release 7**

This document contains information previously presented in LY43-0089-03, which supports z/OS Version 1 Release 6. With release 7, this document becomes an unlicensed publication.

The information in this document includes descriptions of support for both IPv4 and IPv6 networking protocols. Unless explicitly noted, descriptions of IP protocol support concern IPv4. IPv6 support is qualified within the text.

**New information**
- VTAM trace enhancements
  - OON2 entry for module trace (See "OON2 entry for module trace" on page 282.)
- FFST probe ISTFSC3E (See Table 15 on page 435.)
- FFST probe ISTRVM19 (See Table 26 on page 453.)

**Changed information**
- VTAM trace enhancements
  - AFSM entry for altering an FSM state (See "AFSM entry for altering an FSM state" on page 48.)
  - OON entry for module trace (See "OON entry for module trace" on page 281.)
  - OOX entry for module trace (See "OOX entry for module trace" on page 284.)

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You might notice changes in the style and structure of some content in this document–for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

# Chapter 1. Using FFST dumps

This topic covers the FFST dumps that you can use for problem determination for the VTAM program.

See *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for information on other dumps that can be used for problem determination of the VTAM program.

## First Failure Support Technology (FFST) for VTAM

First Failure Support Technology™ is a licensed program that captures information about a potential problem when it occurs. See *z/OS Information Roadmap* to determine what document contains more information on FFST.

When a problem is detected, a software probe is triggered by VTAM. FFST then collects information about the problem and generates output to help solve the problem. Based on the options active for the probe, you get a dump and a generic alert. See "The generic alert" on page 5 for information on generic alerts. You also get the FFST **EPW** message group as shown in the "FFST console" on page 6.

### FFST dumps

Each VTAM FFST probe can trip up to five times in five minutes before it is automatically turned off. Only one of the five dumps will be produced, thereby limiting the number of dumps that you get if a recurring problem triggers a probe.

Depending on how the dump type was coded in the probe and whether or not the VIT data space is present, you can get a full dump, an FFST minidump (partial dump), or both. If the dump type is a full dump, only one full dump is created. If the dump type is a minidump, the FFST minidump is created, and a full dump is also created if the VIT data space is present. For a listing of the dump type for each probe, see Appendix C, "First Failure Support Technology (FFST) probes," on page 429.

If two dumps are created when a minidump probe is triggered, the probe ID can be used as a correlator. The eight-character probe ID is included in the dump title for a full dump and in the primary symptom string for a minidump.

#### Dump data set

FFST saves the VTAM FFST minidump on a dynamically allocated sequential data set. The VTAM FFST full dump is saved on SYS1.DUMPx data sets. You must specify the volume serial number and the UNIT identification information for this data set. Provide this information to FFST on a DD statement in the FFST installation procedure or in the FFST startup command list installed at system installation. A startup command list contains MVS commands to control FFST.

#### Full dump

If a full dump is created when a FFST probe is triggered, FFST uses the operating system SDUMP macroinstruction to provide a full dump of the address space where the potential problem occurred. A full dump includes selected MVS control blocks, CSA, ECSA subpools (227, 228, 231, and 241), the PSA, and the VTAM VIT data space if present.

For more information on data space dumps, see "Recording traces in internal data space table (MODE=INT)" on page 19.

**Formatting a full dump:** Use IPCS to view or print the full dump. If you try to use EPWDMPFM to format a full dump, message `EPW9561E NOT A VALID FFST DUMP` will be issued.

## FFST minidump

If the probe is coded as a minidump, a FFST minidump is written to the output data set. See Figure 1 on page 3.

A FFST minidump contains general purpose registers, selected VTAM control blocks, and the ECSA VIT table.

The probe output data used for VTAM FFST minidumps are found in the data sets that were allocated when VTAM FFST was installed.

**Formatting a FFST minidump:** Use the dump formatting CLIST, EPWDMPFM, to format your VTAM FFST minidump. EPWDMPFM formats your minidump and writes it to a data set that you can view online or print using the IEBPTPCH utility program. (FFST minidumps cannot be processed by the VTAM formatted dump tool.)

**Sample FFST minidump:** See Figure 1 on page 3 for a sample VTAM FFST minidump. Figure 1 on page 3 was produced when VTAM session services CP-CP (SSC) entered an unexpected state on a contention-winner session.

**SCUNO - FAILURE ON CONWINNER SESSION**
11/09/92

---

```
EPW9521I DUMP DATA SET NAME = FFSTDS.MVS42247.VTAM.DMP00033
EPW9522I TITLE FROM DUMP = SCUNO - FAILURE ON CONWINNER SESSION
EPW9523I DATE FROM DUMP = 01/11/95, TIME FROM DUMP = 12:00:06
EPW9501I PRODUCT NAME: VTAM
EPW9502I IBM PROGRAM
EPW9503I COMPONENT/PROGRAM ID: 569511701, LEVEL:  301
EPW9504I TYPE OF FAILURE: INCORROUT
EPW9505I PROBE PRIMARY SYMPTOM STRING:
PIDS/569511701 LVLS/301 PCSS/ISTSCC09 RIDS/ISTSCUNO
EPW9507I REGISTER SECONDARY SYMPTOM STRING:
REGS/GR13 VALU/H068E6098 REGS/GR14 VALU/H868D40CA REGS/GR15 VALU/H00000000
REGS/GR00 VALU/H868D40CA REGS/GR01 VALU/H068E60E0 REGS/GR02 VALU/H068E6200
REGS/GR03 VALU/H0751C000 REGS/GR04 VALU/H86BC5A14 REGS/GR05 VALU/H062D1B88
REGS/GR06 VALU/H00C97EB0 REGS/GR07 VALU/H00C171F8 REGS/GR08 VALU/H068E6078
REGS/GR09 VALU/H00000442 REGS/GR10 VALU/H068D7B60 REGS/GR11 VALU/H00000001
REGS/GR12 VALU/H868D3CAA
0EPW9508I DATA COLLECTION WORK AREA:
0ASID(X'0012') ADDRESS(060B1000) KEY(00)
060B1000. 61004000 00000000 050060F8 01000002 |/. .......-8....|
060B1010. 068E6098 863A27A0 000054A0 06508ED8 |..-qf........&;Q|
060B1020. C5D7E6C4 E2E3E2D2 00000000 068E6020 |EPWDSTSK......-.|
060B1030. 60006200 8652136A 065204E0 00000001 |-...f......\....|
  :
  :
EPW9509I SPECIFIED DATA STRUCTURE TABLE:
ASID(X'0012') ADDRESS(06999BE8) KEY(00)
06999BE8.          77001418 018000F8 |       .......8|
06999BF0. C9E2E3E2 C3C3C6E3 00F9F2F2 F5F64040 |ISTSCCFT.92256  |
06999C00. 0000125C 069711F8 EE0000CC C9D5C3D6 |...*.p.8....INCO|
06999C10. D9D9D6E4 E3404040 002803F8 00003401 |RROUT   ...8....|
  :
  :
EPW9509I DEFAULT DATA STRUCTURE TABLE:
ASID(X'0012') ADDRESS(063859E8) KEY(00)
063859E8.          7700026C 014000F8 |       ...%..8.|
063859F0. C9E2E3D9 C1C3E9E9 E4F9F2F2 F5F64040 |ISTRACZZU92256..|
06385A00. 000001F4 00000000 EE000044 C9D5C3D6 |...4........INCO|
06385A10. D9D9D6E4 E3404040 002803F8 10001C00 |RROUT   ...8....|
  :
  :
```

*Figure 1. Sample FFST minidump (Part 1 of 4)*

```
EPW9510I AREA AROUND REGISTER 0 - 868D40CA:
ASID(X'0012') ADDRESS(068D3CCA) KEY(00)
068D3CCA.                   C8FA 1805140E |          H.....|
068D3CD0. 14E41F0E 55004004 47D0C046 58F0C906 |.U.... ..}{..0I.|
068D3CE0. 5810C902 41000080 89000018 16100A0D |..I.....i.......|
068D3CF0. 182D1838 1E391842 1F550E24 D207D000 |...........K.}.|
:
EPW9510I AREA AROUND REGISTER 15 - 00000000:
ASID(X'0012') ADDRESS(00000000) KEY(00) PREFIXED
00000000. 040C0000 8122F568 00000000 00000000 |....a.5.........|
00000010. 00FD3AC0 00000000 076C2000 826077B4 |...{.....%..b-..|
00000020. 070C1000 800198DE 070C6000 8639E0B2 |......q...-.f.\.|
00000030. 070C1000 8003D52A 070C0000 81E23422 |......N.....aS..|
:
VTAM COMMUNICATION VECTOR TABLE
EPW9520I DATA STRUCTURE DEFINED IN DEFAULT DST
ASID(X'0012') ADDRESS(00C171F8) KEY(00)
00C171F8.                   E5C5F4F3 40404040 |        VE43    |
00C17200. FFF901F4 05F6D000 00000000 0000FFD9 |.9.4.6}.......R.|
00C17210. 11280000 00000000 062FEE88 00000000 |...........h....|
00C17220. 00000000 00000000 00C17524 00000000 |.........A......|
:
```

*Figure 1. Sample FFST minidump (Part 2 of 4)*

```
INTERNAL TRACE TABLE HEADER
EPW9520I DATA STRUCTURE DEFINED IN DEFAULT DST
ASID(X'0012') ADDRESS(05F6D000) KEY(00)
05F6D000. 00000000 00000000 A682A39A 14E4B800 |........wbt..U..|
05F6D010. A682A398 2717CF00 05F6F040 05F9EFE0 |wbtq.....60 .9.\|
INTERNAL TRACE TABLE
EPW9514I EXIT ROUTINE CALLED FOR DATA STRUCTURE
EPW9520I DATA STRUCTURE DEFINED IN DEFAULT DST
ASID(X'0012') ADDRESS(05F6F060) KEY(00)
05F6F060. C6C2D3D2 12001200 069DFF20 86387AFC |FBLK........f.:.|
05F6F070. 868BBD02 00000020 00000000 0100FFFF |f..............|
05F6F080. C7C2D3D2 12001200 06F46E78 06387B40 |GBLK.....4>...# |
05F6F090. 868BCE08 00000040 00000000 00000031 |f...... ........|
:
CONTINUATION OF INTERNAL TRACE TABLE
EPW9514I EXIT ROUTINE CALLED FOR DATA STRUCTURE
EPW9520I DATA STRUCTURE DEFINED IN DEFAULT DST
ASID(X'0012') ADDRESS(05F6D020) KEY(00)
05F6D020. D7C9E440 12990000 06368CF8 40000902 |PIU .r.....8 ...|
05F6D030. 200007FC 000001F4 00000136 1C000001 |.......4........|
05F6D040. D7C9E4F2 0306010A 00100380 00C4C1E3 |PIU2.........DAT|
05F6D050. C140C1F3 F1C4F9F2 C1F50000 00000000 |A A31D92A5......|
:
```

*Figure 1. Sample FFST minidump (Part 3 of 4)*

```
   SCDAT - SSC PROCESS DATA
   ASID(X'0012') ADDRESS(062D1B88) KEY(00)
   062D1B88.                   E2C3C4E3 00000000 |       SCDT....|
   062D1B90. C0000088 06A37020 06A37138 06B259F8 |{..h.t...t.....8|
   062D1BA0. 00041100 00000000 00000000 00000000 |................|
   062D1BB0. 00000000 00000000 06543018 00000000 |................|
   062D1BC0. 00000000 0751C000 00000000 06A37020 |......{......t..|
   062D1BD0. 60C3D7E2 E5C3D4C7 40000000 00000000 |-CPSVCMG .......|
   062D1BE0. 000C12C1 00000000 F6BE0000 00000000 |...A....6.......|
   062D1BF0 LENGTH(1040)==>All bytes contain X'00'
   RUPE - REQUEST UNIT PROCESSING ELEMENT
   ASID(X'0012') ADDRESS(0751C000) KEY(00)
   0751C000. 54136004 00000000 00000000 00000000 |..-............|
   0751C010. 00000000 00000000 00000000 FDC3D7E2 |............CPS|
   0751C020. 00000000 D5C5E3C1 4BC1F8F1 D5404040 |....NETA.A81N  |
   0751C030. 40404040 40404040 40404040 40000000 |            ...|
   0751C040 LENGTH(16)==>All bytes contain X'00'
   0751C050. 068E5E08 00000000 40060000 00004004 |..;.... ..... .|
   0751C060. 00000000 00000000 00000000 86BC5A14 |............f.!.|
   0751C070. 00000000 081C0002 00000000 00000000 |................|
   0751C080 LENGTH(16)==>All bytes contain X'00'
   0751C090. 00000000                            |....           |
   IPS - INTER PROCESS SIGNAL
   ASID(X'0012') ADDRESS(06BC5A14) KEY(00)
   06BC5A14.          0440FDC3 D7E202D5 C5E3C14B |  . .CPS.NETA.|
   06BC5A20. C1F8F1D5 40404040 40404040 00000000 |A81N        ....|
   06BC5A30. 00090000 00000300 0005081C 00021400 |................|
   06BC5A40 LENGTH(1040)==>All bytes contain X'00'
   06BC5E50. 00000000 0000                       |......         |
   ANDCB - ADJACENT NODE CONTROL BLOCK
   0ASID(X'0012') ADDRESS(06A37020) KEY(00)
   06A37020. C1D5C3C2 D5C5E3C1 4BC1F8F1 D5404040 |ANCBNETA.A81N  |
   06A37030. 40404040 40000000 00000009 C3D7E2E5 |     .......CPSV|
   06A37040. C3D4C740 00000000 00000000 00000000 |CMG ............|
   06A37050 LENGTH(80)==>All bytes contain X'00'
   06A370A0. 00000000 02000000 00000000 06B62138 |................|
   06A370B0 LENGTH(16)==>All bytes contain X'00'
   06A370C0. 0751BE20 E0000000 02000000 00000000 |....\..........|
   06A370D0. 0751BEC0 00000000 00000000 80000000 |...{............|
   06A370E0. 00000000 10F01002 02000000 0751C1E0 |.....0........A\|
   06A370F0. 01000000 00000000 00000000 00000000 |................|
   06A37100. 00000000 00000000 00000000 A0801010 |................|
   06A37110. 00000000 00000000 06BB9608 00000000 |..........o.....|
   06A37120. 00000000 00000000 00000000 00       |............   |
```

*Figure 1. Sample FFST minidump (Part 4 of 4)*

# The generic alert

A software generic alert is built from the symptom record and routed to the NetView® program if installed. The generic alert contains:

- The date and time that the probe was triggered
- The system name from the CVTSNAME field
- The product name (VTAM)
- The component identification and release number of the product triggering the probe
- The hardware identification information:
  - Machine type
  - Serial number
  - Model number
  - Plant code
- The dump data set and volume if a dump was taken

- The probe statement identifier
- The probe statement description
- The probe statement severity level

## The symptom string

The primary symptom string contains the following data supplied by VTAM:
- PIDS/component ID: The VTAM component identifier
- LVLS/level: The VTAM specification for the product level
- PCSS/Probe ID: From the probe that was triggered
- PCSS/FULL or MINI: The type of dump taken
- RIDS: Module name from the probe that was triggered

## FFST console

See Figure 2 for a sample console listing for FFST. In Figure 2 the FFST program console message group **EPW** shown informs you that a probe has been triggered and that data is being collected. The **EPW0404I** messages contain the primary symptom string for VTAM.

```
EPW0401I FFSTPROC: ERROR DETECTION INVOKED BY VTAM 287
EPW0406I DUMP DATASET IS: USER1.SP41D23.VTAM.DMP00002
EPW0407I FOUND ON VOLUME: CPDLB2
EPW0402I PRIMARY SYMPTOM STRING FOR VTAM FOLLOWS:
EPW0404I PIDS/569511701 LVLS/301 PCSS/ISTTSC01 PCSS/VR#HANG PCSS/MINI
EPW0404I RIDS/ISTTSCRI FLDS/TH4VRSSN VALU/H0000 FLDS/VRBSQRCV
EPW0404I VALU/H0000 FLDS/VRBDSTSA VALU/H00000000
EPW0701I END OF MESSAGE GROUP
F FFSTPROC,AP=FFST
EPW0610I FFSTPROC: DISPLAY FOR APPLID FFST FOLLOWS: 294
EPW0611I APPLID COUNTS    probe  DUMP SYMRC GENAL SYMST SUPDP
EPW0612I FFST   00002/00002 EN   EN   EN    EN    EN    EN
EPW0613I DUMPQUAL = USER1, DUMPVOL = CPDLB2
EPW0614I APPLID VENDOR
EPW0615I VTAM  IBM CORPORATION
EPW0701I END OF MESSAGE GROUP
```

*Figure 2. Sample VTAM FFST console listing*

## Using the trap module

The IBM-supplied trap module (ISTRACZT) contains pretested probes that you can use to capture data in places where a probe has not been installed inside of VTAM.

### Steps for using the trap module

**Before you begin:** You need to obtain the VTAM module name and offset in the VTAM module from IBM Service.

Perform the following steps to install the trap program:

1. Add an instruction to check the ATCFFST field in the ISTATCVT control block. If ATCFFST is 0, VTAM FFST is not available.

   _____

2. Add an instruction to check the ATCRACZT field in the ISTATCVT control block. If ATCRACZT is 0, the trap module is not available.

   _____

3. Add an instruction to call the trap module.
   ```
   BALR R14,R15 -------- 05EF
   DC   X'0001' -------- 0001 (default)
   ```

A 2-byte field containing X'0001' or X'0002' follows the BALR instruction. A value of X'0001' issues the ISTRAC01 probe macro in VTAM with the SDUMP option. An index value of X'0002' issues the ISTRAC01 probe macroinstruction with the VTAM FFST minidump option. The ISTATCVT and the VIT are included in the minidump output along with the VTAM module list, which contains the five significant letters of a module name, its service level, and its address. Control is returned to VTAM at the address following the 2-byte index.

_____

You know you are done when you execute the program and the trap is triggered, resulting in an SDUMP or FFST minidump (depending on the option chosen).

**Notes:**

1. The calling module must save GP register 7.
2. The module is reentrant.
3. AMODE is 24 or 31.
4. All registers except 7 are saved and restored.
5. The module is in LPALIB.
6. If you are trying to invoke the trap module outside of the VTAM environment, you will need to do the following:
   - Follow the list shown above.
   - Be in VTAM key (6).
   - Be authorized.

# When to dump coupling facility structures

When using GR, MNPS, TSO/GR, TCP/IP Sysplexports, or TCP/IP Sysplex Wide Security Associations be sure to dump the coupling facility structures involved when documenting problems with those functions. Refer to *z/OS MVS System Commands* for information on dumping a coupling facility structure.

# APPC sense code trap

The IBM-supplied trap is placed in the VTAM APPC component. This trap contains a pretested probe that provides a full dump when a predetermined sense code is set. The sense code will trigger probe ISTRACZ3.

## Steps for activating an APPC sense code trap

**Before you begin:** You need to determine the sense code to be trapped. Sense codes are listed in *z/OS Communications Server: IP and SNA Codes*.

Perform the following steps to activate a trap:

1. Ensure that the FFST program is operational.

   _____

2. Ensure that the APPC VIT option is active.

   _____

3. Obtain the offset of the ATCASLIP field in the ATCVT. (This field is a full word.)

   _____

4. If you are running an MVS guest on a VM system, using the CP TRACE STORE command, set ATCASLIP to the desired sense code.

   _____

You know you are done when the FFST probe ISTRACZ3 is triggered.

## Phantom First Failure Support Technology (PFFST) for VTAM or CSM

Phantom First Failure Support Technology (PFFST) captures information about a potential problem when it occurs if FFST is not installed or active. Refer to *z/OS Information Roadmap* to determine what document contains more information on FFST.

Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for information on other dumps that can be used for problem determination of the VTAM program or CSM.

When a problem is detected, a software probe is triggered by VTAM or CSM. If FFST is not installed or active, then PFFST collects information about the problem and generates output to help solve the problem.

## Reporting a problem

Some of the probes that are triggered are not VTAM problems. Analyze the probe output to determine if a VTAM problem exists before you contact an IBM Support Center representative. For more information on FFST probes, see Appendix C, "First Failure Support Technology (FFST) probes," on page 429.

Some probes might be triggered by VTAM problems that have already been fixed. To determine whether a problem has already been solved, take the following steps:

1. Search the RETAIN® database for occurrences of the symptom string you receive when the probe is triggered. [Also search the Information System (I/S) database or the database used by your organization.]
2. If you find an APAR that applies, apply the fix.
3. If you do not find an APAR and you cannot fix the problem, report it.

   For non-VTAM problems, call your IBM branch office. For suspected VTAM problems, do either of the following steps:
   - Access IBMLink™ and search for a similar problem by using the symptom string. If no matches are found, report the problem to IBM by using the electronic technical report (ETR) option on IBMLink.
   - Contact the IBM Software Support Center at 1-800-IBM-SERV.

If you call the IBM Support Center, the Center must verify that the documentation collected is adequate to fix the problem and that the problem is a VTAM problem.

If the problem is a VTAM problem, the IBM Support Center opens an APAR against VTAM and includes the symptom string generated by the probe as part of the APAR text.

If the problem is a hardware, network definition, or user definition error, the IBM Support Center representative creates an ASKQ item for VTAM. The ASKQ item includes the symptom string and the solution for the problem, and can be found in the problem determination database (PDDB).

# Chapter 2. Using the VTAM internal trace (VIT)

This topic includes the following information:

- "Activating the internal trace"
  - "Selecting trace options" on page 10
  - "Internal and external trace recording" on page 17
  - "Module names in internal trace records" on page 21
  - "SNAP trace record" on page 23
- "Deactivating the VTAM internal trace" on page 25

Most VTAM traces show the information flow between the VTAM program and other network components. However, the VTAM internal trace (VIT) provides a record of the sequence of events *within* VTAM. These internal events include the scheduling of processes (for example, POST, WAIT, and DISPATCH), the management of storage (for example, VTALLOC), and the flow of internal PIUs between VTAM components.

Together with the operator console listing and a dump, output from the VIT can help you reconstruct sequences of VTAM events and find internal VTAM problems more easily.

## Activating the internal trace

Trace data for the following VIT options is always automatically recorded in the internal table:

- API
- CIO
- MSG
- NRM
- PIU
- PSS (may be turned off to stop automatic recording)
- SMS (may be turned off to stop automatic recording)
- SSCP

Use one of the following methods to start the VIT:

- You can use the TRACE start option, with TYPE=VTAM specified, to start the VIT when you first start VTAM. See *z/OS Communications Server: SNA Resource Definition Reference* for more information on how to use the TRACE start option.

- You can use the MODIFY TRACE command, with TYPE=VTAM specified, to start the VIT after you have started VTAM. See *z/OS Communications Server: SNA Operation* for more information on how to use the MODIFY TRACE command.

**Notes:**

1. To prevent the VIT table from being overwritten, VTAM disables the internal VIT when it issues SDUMP and when a FFST probe is tripped.

2. The minimum trace table size is 100 pages. Because the eight trace option defaults are always running, the table may wrap many times.

3. CIDCTL FIND macro invocations executed during the process of sending or receiving data will not be traced with CDHF or CDNF trace entries unless they result in a nonzero return code.

## Selecting trace options

Both the TRACE start option and the MODIFY TRACE command have an
OPTION operand you can use to select VIT options. Select one or more of the
following options to indicate the VTAM functions you want to trace.

**Note:** If you do not deactivate the VIT before you attempt to change an option, the
options that are currently in effect will remain in effect. See "Deactivating
the VTAM internal trace" on page 25 for more information on deactivating
the VIT.

- **API** option (for application programming interfaces)

  This option helps you determine whether an application program is causing a
  problem. API entries are written for RPL macros, RPL exit routines, user exit
  routines, and user posts.

  Trace data for this option is always automatically recorded in the internal table.

- **APIOPTS** option

  This option is a collection of multiple VIT options that includes all of the
  individual VIT options usually required to diagnose potential application
  program problems. Specifying the APIOPTS option is equivalent to specifying all
  of the following VIT options: API, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **APPC** option (for LU 6.2 application programming interfaces)

  This option helps you determine whether an LU 6.2 application is causing a
  problem. LU 6.2 entries are written for APPCCMD macro invocations, user posts
  and exit scheduling by LU 6.2 code, calls to a security manager for security
  processing, and message unit transmissions between LU 6.2 components.

- **APPCOPTS** option

  This option is a collection of multiple VIT options that includes all of the
  individual VIT options usually required to diagnose potential LU 6.2 application
  program problems. Specifying the APPCOPTS option is equivalent to specifying
  all of the following VIT options: API, APPC, MSG, NRM, PIU, PSS, SMS, and
  SSCP.

- **CFS** option (for coupling facility interfaces)

  This option helps you determine problems with the VTAM interface with the
  MVS coupling facility. CFS entries are written when VTAM issues MVS macros
  to request coupling facility related services.

- **CIA** option (for channel input and output auxiliary)

  This option helps you isolate problems related to channel I/O CIA entries. This
  option presents the remaining trace records from the CIO option.

- **CIO** option (for channel input and output)

  This option helps you isolate problems related to channel I/O. CIO entries are
  written for attentions, error recovery, interruptions, HALT I/O SVC, and START
  I/O SVC.

- **CMIP** option (for Common Management Information Protocol Services)

  Setting the CMIP option traces:

  - Calls from CMIP application programs to the management information base
    (MIB) application programming interface
  - Calls to the read-queue exit of the CMIP application program
  - Topology updates from VTAM resources

  You can use the CMIP option to help you determine whether there is a problem
  in VTAM or in a CMIP application program.

- **CPCPOPTS** option

This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose potential CP-CP session problems. Specifying the CPCPOPTS option is equivalent to specifying all of the following VIT options: API, APPC, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **CSM** option (for communications storage manager events)

This option traces the parameter list information that flows across the CSM interface and key internal events (such as pool expansion and contraction) for functions that manipulate buffer states. This allows you to trace and analyze the usage history of a buffer.

You can also use the CSM trace when VTAM is not operational. An external trace is generated using the VTAM GTF event ID to write trace records directly to GTF in the same format as those recorded using VIT.

- **CSMOPTS** option

This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose potential communications storage manager (CSM) problems. Specifying the CSMOPTS option is equivalent to specifying all of the following VIT options: API, APPC, CIO, CSM, MSG, NRM, PIU, PSS, SMS, SSCP, and XBUF.

- **DLUROPTS** option

This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose dependent LU requester (DLUR) problems. Specifying the DLUROPTS option is equivalent to specifying all of the following VIT options: API, APPC, HPR, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **EEOPTS** option

This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose Enterprise Extender (EE) problems. Specifying the EEOPTS option is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP, and TCP.

- **ESC** option (for execution sequence control)

This option helps you track in detail the flow of requests for a given process.

- **HPDTOPTS** option

This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose high-performance data transfer (HPDT) problems. Specifying the HPDTOPTS option is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, PIU, PSS, SMS, and SSCP.

- **HPR** option (for High-Performance Routing)

This option helps you isolate problems related to High-Performance Routing.

- **HPROPTS** option

This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose High-Performance Routing (HPR) problems. Specifying the HPROPTS option is equivalent to specifying all of the following VIT options: API, APPC, CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **LCS** option (for local area network (LAN) channel stations)

This option helps you isolate problems occurring during activation of, deactivation of, and data transfer from an IBM 3172 Interconnect Nways® Controller. The LCS option enables tracing of data that VTAM receives from an

IBM 3172 Interconnect Nways Controller at four levels: LCSX (channel), LCSP (port or adapter), LCSS (SAP), and LCSL (line).

- **LCSOPTS** options

  This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose LAN channel station (LCS) problems. Specifying the LCSOPTS option is equivalent to specifying all of the following VIT options: CIO, LCS, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **LOCK** option (for locking and unlocking)

  This option helps you determine when VTAM modules get and release locks.

- **MSG** option (for messages)

  This option helps you:
  - Correlate other VIT entries with the console messages even if you lose the console sheet. MSG entries are written for all messages to the VTAM operator.
  - Match the console log to a surge of activity shown in the VIT. OPER entries are written for all VTAM commands issued at an operator console.

  Trace data for this option is always automatically recorded in the internal table.

- **NRM** option (for network resource management)

  This option helps you follow the services of the network resource management component. These include the assignment of, references to, and the deletion of certain VTAM resources such as node names, network addresses, and control blocks. NRM entries are written for SRT macros issued by VTAM modules.

  Trace data for this option is always automatically recorded in the internal table.

  CIDCTL FIND macro invocations executed during the process of sending or receiving data will not be traced with CDHF or CDNF trace entries unless they result in a nonzero return code.

- **PIU** option (for path information unit flows)

  This option, like the I/O and buffer contents traces, helps you isolate problems to hardware, to the NCP, or to VTAM. Unlike I/O and buffer contents traces, with this option PIU entries are written for all PIUs that flow internal and external to VTAM.

  Trace data for this option is always automatically recorded in the internal table.

- **PSS** option (for process scheduling services)

  This option helps you track the flow of requests through VTAM. PSS entries are written for the VTAM macros that invoke and control PSS, scheduling and dispatching VTAM routines.

- **QDIOOPTS** options

  This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose queued direct I/O (QDIO) problems. Specifying the QDIOOPTS option is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **SMS** option (for storage management services)

  This option helps you isolate problems caused by storage shortages. When used with the SSCP or PSS trace options, it can also help you isolate internal VTAM problems. SMS entries are written when SMS macros are used to request or free fixed-length or variable-length buffers. SMS entries are also written when VTAM expands or attempts to expand a buffer pool.

- **SSCP** option (for system services control point request scheduling and response posting)

This option helps you isolate a VTAM problem to a specific VTAM component or module. SSCP entries are written for the request/response units (RUs) sent between VTAM components. This option also records information for the APPN CP.

Trace data for this option is always automatically recorded in the internal table.

- **STDOPTS** option

  This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose problems related to high CPU, session services, storage, Open/Close ACB, and DLCs such as multipath channel (MPC) and channel-to-channel (CTC). Specifying the STDOPTS option is equivalent to specifying all of the following VIT options: API, CIO, MSG, NRM, PIU, PSS, SMS, and SSCP.

- **TCP** option (for use with Enterprise Extender)

  This option is used for recording activity related to Enterprise Extender. The trace options record IP address management as well as timer activity.

- **TCPOPTS** option

  This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose problems related to TCP/IP. Specifying the TCPOPTS option is equivalent to specifying all of the following VIT options: CIA, CIO, MSG, NRM, PIU, PSS, SMS, SSCP, and TCP.

- **VCNS** option (for VCNS application programming interfaces)

  This option helps you determine whether a VCNS application is causing a problem. VCNS entries are written for VCNSCMD macro invocations, user posts, and exit scheduling by VCNS code, and work element transmissions between VCNS components.

- **XBUF** option (for applications using the extended buffer list for sending and receiving data)

  This option traces the contents of the extended buffer list (XBUFLST). Records are produced to trace these contents from the application supplied extended buffer list as well as the internal buffer list that VTAM uses to carry the extended buffer list information. These records store relevant information contained with the extended buffer list, particularly information on CSM usage by VTAM.

- **XCF** option (for VTAM use of the cross-system coupling facility)

  This option allows you to track VTAM use of the XCF (cross-system coupling facility) MVS macro interface. There is a VIT entry for each VTAM use of an XCF macro.

- **XCFOPTS** option

  This option is a collection of multiple VIT options that includes all of the individual VIT options usually required to diagnose cross-system coupling facility (XCF) problems. Specifying the XCFOPTS option is equivalent to specifying all of the following VIT options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP, and XCF.

The VIT always traces the exception conditions listed in Table 1 on page 14 and all the default VIT options listed under "Activating the internal trace" on page 9.

## Activating the VIT

*Table 1. Exception conditions always traced by the VTAM internal trace*

| Option | Exception conditions traced |
|---|---|
| APPC | • ACA and ACI entries when issuing the following commands:<br>  – SEND ERROR<br>  – DEALLOC ABNDxxxx<br>  – REJECT<br>• ACRC and ACSN entries<br>• Other entries with nonzero return codes (except RPL6RCSC) |
| CFS | Entries with nonzero return codes |
| CIO | INOP entry |
| CMIP option | The following entries, when they have nonzero return codes:<br>• MCO1 and MCO2<br>• MDEL<br>• MDIS<br>• MQRQ<br>• MQRS<br>• MREG<br>• RQE |
| LCS | LCSL, LCSP, LCSS, and LCSX entries with nonzero reason codes |
| NRM | CDHF or CDNF entries with nonzero return codes |
| SMS | Entries with nonzero return codes and EXPN entries if a buffer pool expansion fails |
| SSCP | CPI, CPO, and CP2 |
| (No option) | All SNAP entries and some exception entries (see 1 on page 16). |

*Table 2. VIT options and the records they create (API - LOCK)*

| VIT options | API | APPC | CFS | CIA | CIO | CMIP | CSM | ESC | HPR | LCS | LOCK |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **VIT records** | **AIx** | ACAx | CFAx | DEVx | **ADE** | MCO1 | ASNx | ESC | ARB | LCSx | LKEX |
| | **IOx** | ACIx | CFCx | ENFx | **ATT** | MCO2 | CHGx | | ARBB | | LKSH |
| | **RE** | ACPx | CFDx | GCEL | **ERPx** | MDEL | CNTP | | ARBR | | ULKA |
| | **UEx** | ACRx | CFEx | GCEx | **HIOx** | MDIS | CPYx | | ARPx | | UNLK |
| | **UP** | ACSN | CFFC | IDx | **INTx** | MQRQ | EXPP | | ARQx | | |
| | | ACUx | CFLx | IUTx | **PCIx** | MQRS | FIXx | | ARSx | | |
| | | MUx | CFNF | LNKx | **RIOx** | MREG | FRBx | | DAPT | | |
| | | RACR | CFPx | LSNx | **SIGA** | MRGx | GTBx | | DRPx | | |
| | | REML | CFRB | MPDx | **SIOx** | RQE | PAGx | | HCLK | | |
| | | REMQ | CFTx | ODPx | | | | | HPRx | | |
| | | USx | CFUS | ODTx | | | | | HPRT | | |
| | | UVx | CFVC | PKx | | | | | NLPx | | |
| | | | MNPS | PLOx | | | | | ONLP | | |
| | | | | SBAx | | | | | OOSx | | |
| | | | | SLSx | | | | | RCM | | |
| | | | | TOKx | | | | | RCV | | |
| | | | | XIDx | | | | | REML | | |
| | | | | | | | | | RSCx | | |
| | | | | | | | | | RTP | | |
| | | | | | | | | | RTPx | | |
| | | | | | | | | | RTSx | | |
| | | | | | | | | | RVM | | |
| | | | | | | | | | RXMT | | |

*Table 3. VIT options and the records they create (MSG - XCF)*

| VIT options | MSG | NRM | PIU | PSS | SMS | SSCP | TCP | VCNS | XBUF | XCF |
|---|---|---|---|---|---|---|---|---|---|---|
| **VIT records** | MSGx<br>OPEx<br>QRYL<br>TRNx | BSPx<br>BSSx<br>BSXx<br>CDHx<br>CDNx<br>NIPx<br>PROx<br>RCEx<br>SRTx | DCOx<br>DSCx<br>NRSx<br>PIUx<br>RDSx<br>TSNS | ATSK<br>BTSK<br>DSP<br>DTSK<br>ETSK<br>EXIT<br>IRBx<br>POST<br>QUEx<br>RESM<br>SCHD<br>SRBx<br>VPST<br>VRSM<br>VWAI<br>WAIT<br>XPST | AREL<br>CONT<br>EXPN<br>FBLx<br>FRES<br>GBLx<br>GETS<br>ORMG<br>POOF<br>QREx<br>RELS<br>REQx<br>VTAL<br>VTFR | AFSM<br>ALSx<br>AP<br>A2<br>CCx<br>CIx<br>COx<br>CPI<br>CPO<br>CP2<br>CPPx<br>CPRx<br>CPWx<br>CRx<br>CSx<br>DBx<br>DLTx<br>ENR<br>GNAx<br>HLSx<br>LDLx<br>MT<br>SPTx<br>TGMx<br>TGVx<br>TOPx<br>TPN2<br>TPTx<br>TREx<br>TRMx<br>TRRx | IPAD<br>IPGN<br>IPG2<br>IPG3<br>IPOG<br>IPO2<br>IPTC<br>IPTM | CNA<br>CNPx<br>CNRx<br>NSD<br>VCCx<br>VCDQ | XBAx<br>XBIx | XCC2<br>XCFC<br>XCFJ<br>XCFL<br>XCFM<br>XCFR<br>XCFS<br>XCFX<br>XCJ2<br>XCL2<br>XCM2<br>XCR2<br>XCS2 |

**Notes:**

1. The **** (FFST and PFFST), ABND, BUFF, COPY, CMER, CME2, INOP, LOST, MMG, and MM2 trace records are not activated by specific VIT options. They are activated as a result of exception conditions.

2. Most of the entries in boldface type are forced, that is, always traced even if they are not specified by the user. However, the entries under SMS and PSS can be turned off.

3. For CIO record types ATT, ERP, HIO, INT, SIO, with suffix I, X, or T, and INOP, the events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

4. OON and OOX can be generated when the module trace is running.

5. For the IRBx and the SRBx records to be recorded, both the PSS trace option and the PSSTRACE start options must be specified.

6. For APPC record types REMQ and ACSN, the events are also captured within the ISTRAB.

7. Some trace records are generated only when a subtrace is active. These trace records are the HPR option record types ARBB, ARBR, and the SSCP option record types HLSx, TGVx, TRMx, and TRRx. For more information about subtraces, refer to *z/OS Communications Server: SNA Operation*.

Table 4 shows the VIT group options and the individual VIT option equivalent for each group option.

*Table 4. VIT group options*

| VIT group option | Equivalent to this set of individual VIT options |
| --- | --- |
| APIOPTS | API, MSG, NRM, PIU, PSS, SMS, SSCP |
| APPCOPTS | API, APPC, MSG, NRM, PIU, PSS, SMS, SSCP |
| CPCPOPTS | API, APPC, MSG, NRM, PIU, PSS, SMS, SSCP |
| CSMOPTS | API, APPC, CIO, CSM, MSG, NRM, PIU, PSS, SMS, SSCP, XBUF |
| DLUROPTS | API, APPC, HPR, MSG, NRM, PIU, PSS, SMS, SSCP |
| EEOPTS | CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP, TCP |
| HPDTOPTS | CIA, CIO, HPR, MSG, PIU, PSS, SMS, SSCP |
| HPROPTS | API, APPC, CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP |
| LCSOPTS | CIO, LCS, MSG, NRM, PIU, PSS, SMS, SSCP |
| QDIOOPTS | CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP |
| STDOPTS | API, CIO, MSG, NRM, PIU, PSS, SMS, SSCP |
| TCPOPTS | CIA, CIO, MSG, NRM, PIU, PSS, SMS, SSCP, TCP |
| XCFOPTS | CIA, CIO, HPR, MSG, NRM, PIU, PSS, SMS, SSCP, XCF |

# Internal and external trace recording

VTAM can write the VIT trace data to an internal table or an external device, such as a disk or tape. You specify internal or external with the MODE operand of the TRACE start option or the MODIFY TRACE command. The VIT record contains the same information regardless of the MODE selected.

You can record data externally and internally at the same time, and if desired, you can have different sets of trace options active for each mode. The default trace options (API, CIO, MSG, NRM, PIU, PSS, SMS, and SSCP) are always recorded internally; however, PSS and SMS can be turned off.

### Recording traces in internal CSA table (MODE=INT)

If you set MODE=INT on the MODIFY TRACE command or as a TRACE start option, or if you let MODE default to INT, VTAM writes the VIT trace records in an internal trace table. The table is allocated and initialized in extended common service area (CSA) storage.

The SIZE operand of the TRACE start option specifies the number of pages (1–999) in storage to be allocated for the internal trace table. Each page is 4K. If you omit this option, the default size is 100 pages. If you specify fewer than 100 pages, VTAM uses 100. Because it is a wraparound table, specify enough pages to ensure that the VIT will not overwrite important trace records when the table fills and begins to wrap around.

The SIZE operand should also reflect the amount of storage available for the internal trace table. If there is not enough storage available for the number of pages specified, you will receive a message indicating that internal trace activation failed.

After the VIT is started, the size operand does not have a default. You can change the size of the internal trace table by issuing a MODIFY TRACE command with a

new SIZE operand. However, if you change the table size while the VIT is running, the current internal trace table is freed, and VTAM starts a new one. Therefore, the trace information in the current table will be lost.

Figure 3 shows an example of VTAM internal trace records written with MODE=INT and printed in dump output. Six of the eight default trace options (API, CIO, MSG, NRM, PIU, and SSCP) are always active.

```
E2C3C8C4 14202810 06308E88 00C17248  8652D180 00000000 C9D5E3D4 00000000  *  SCHD.......h.A..f.J.....INTM....  *
E2D9C2C4 14000000 06308E88 80000000  00000000 00000000 007FF158 00800000  *  SRBD.......h.............."1.....  *
E2D9C2E7 14170000 06308E88 00000000  00000000 00EBDE80 00EBDE80 824C4B82  *  SRBX.......h.................b<.b  *
C9D9C2C4 14000000 06308E88 80000000  00C17248 007DDB18 007FF158 00800000  *  IRBD.......h......A...'..."1.....  *
D9C5D8E2 14170000 06308E88 06324810  81C23E1A 00010000 06300DA0 00000000  *  REQS.......h....aB..............  *
C4E2D740 1400A810 06308E88 00C17248  00000000 00000000 C9D5E3D4 06324810  *  DSP ..y....h.A..........INTM....  *
C3C3C900 14588000 06324A20 00000000  8652D13A 00000000 58588000 00000000  *  CCI.......+.....f.J.............  *
D8E4C558 14482410 06308E88 00C17908  824EAA8C 0651C928 E3E2E6E4 06324810  *  QUE........h.A..b+....I.TSWU....  *
C5E7C9E3 14000010 06308E88 00C17248  8652CF74 80000000 C9D5E3D4 06324810  *  EXIT.......h.A..f.......INTM....  *
D9C5D3E2 14170000 06308E88 06324810  81C23E92 00000000 06300DA0 00000000  *  RELS.......h.....aB.k...........  *
C4E2D740 14582410 06308E88 00C17908  0651C928 0651C928 E3E2E6E4 0631D810  *  DSP .......h.A....I...I.TSWU..Q.  *
D8E4C558 14482810 06308E88 00C17248  86615658 0651C928 C9D5E3D4 0631D810  *  QUE........h.A..f/....I.INTM..Q.  *
C5E7C9E3 14000010 06308E88 00C17908  86615708 80000000 E3E2E6E4 0631D810  *  EXIT.......h.A..f/......TSWU..Q.  *
D9C5D8E2 14170000 06308E88 06324810  81C23E1A 00010000 06300DA0 00000000  *  REQS.......h....aB..............  *
C4E2D740 14582810 06308E88 00C17248  0651C928 0651C928 C9D5E3D4 06324810  *  DSP .......h.A....I...I.INTM....  *
C5E7C9E3 14000010 06308E88 00C17248  8652CF74 80000000 C9D5E3D4 06324810  *  EXIT.......h.A..f.......INTM....  *
```

*Figure 3. Unformatted VIT records in dump output (MODE=INT)*

The first line of the internal trace table is the header. (The header is not included in the preceding example.) The header contains the following status information:

**Byte (hex)**
> **Contents**

**00–07**  Zero
**08–0F**  Present wraparound time stamp
**10–17**  Last wraparound time stamp
**18–1B**  Address of the most recent entry in the table
**1C–1F**  Address of the last entry in the table

**Dump analysis:**  Several dump analysis tools are available for the VTAM internal trace:

- VITAL extracts an internal VIT from a dump for use with the VIT analysis tool.
- VTBASIC displays the VIT table.
- VTVIT displays the VIT options that were in effect at the time of the dump and whether the trace was running internally, externally, or both.

See *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for more information on the VTAM dump analysis tools.

Use the procedure that follows as an alternative to the dump analysis tools.

Examine the internal trace table by taking a dump of VTAM with the appropriate storage area. Find the internal trace table in the dump by first locating the ATCVT. The pointer to the ATCVT (ATCLCPTR) is at low-storage location X'408'.

If this low-storage location is not available in a dump, use the pointer in the MVS CVT control block extension (CVTATCVT) to find the VTAM control block AVT. Location hex 00 in the AVT points to the ATCVT. In the ATCVT, field ATCITTBL contains a pointer to the trace table. ATCITTBL is located at offset X'C'. The internal trace table contains a 32-byte header followed by 32-byte trace records.

## Recording traces in internal data space table (MODE=INT)

To allow VTAM to extend the size of the internal trace table, VTAM also provides the capability of tracing to a data space. The size of this data space is controlled with the DSPSIZE (data space size) operand of the MODIFY TRACE command. The range is 1 to 5 (10 to 50 megabytes). Note that the largest CSA table (999 pages) is approximately four megabytes.

When tracing to the data space, the CSA table is still used (minimum of 100 pages). Therefore, when tracing to the data space, keep the CSA size (SIZE=*operand*) small (100 to 200 pages).

The size of the data space table can also be changed while tracing, but the same exposure exists (as with the CSA table). The current table is freed, and VTAM starts a new data space table.

The format of the records within the data space are the same as the CSA table. The tools to extract trace entries in the data space table are all the same as in the CSA table. The VTAMMAP VTVIT, VTBASIC, and VITAL functions will now process the VIT entries in the data space in addition to the VIT entries in the table in fixed ECSA in primary storage.

The most recent VIT entries are in the fixed ECSA table. Periodically, the VIT entries are copied to the data space. It is very likely that some VIT entries will be duplicated in the fixed ECSA table and in the data space. To unravel the VIT, do the following:

- See Figure 4. First, look in the VIT in fixed ECSA. The most recently written entry is at ITHCURR. From there, work up to ATCITTBL (1). (ATCITTBL actually points to the header, so begin at offset X'20'.) For the next most recent entries (2), start from ITHLAST and work up to ITHCURR+X'20'.
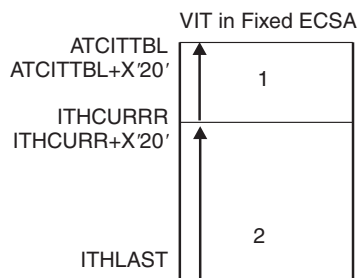


*Figure 4. VIT in fixed ECSA*

- See Figure 5 on page 20. In the data space, the most recently written entries are from DSDCB_IT_NEXT up, but some of these are duplicates of the VIT in fixed ECSA. Use the VTAMMAP VTVIT command; DVITC is the newest nonduplicate VIT entry. If the data space has wrapped (DSDCB_IT_WRAPPED) (4), continue reading from the last entry in the data space (DSDCB_IT_LAST) to DVITC.

VIT in data space

```
DSDDATA ┌──────────┐   First entry in data space
        │    ↑     │   table
        │    3     │
        │          │   When unwrapping the VIT in a
  DVITC ├──────────┤   dump,
        │          │       you can look at all the
        │          │   entries
        │          │       in the CSA VIT. Then start
DSDCB_IT_NEXT ├────┤   here
        │    ↑     │       and look up (3) for next
        │    4     │   entries.
DSDCB_IT_LAST └────┘
```
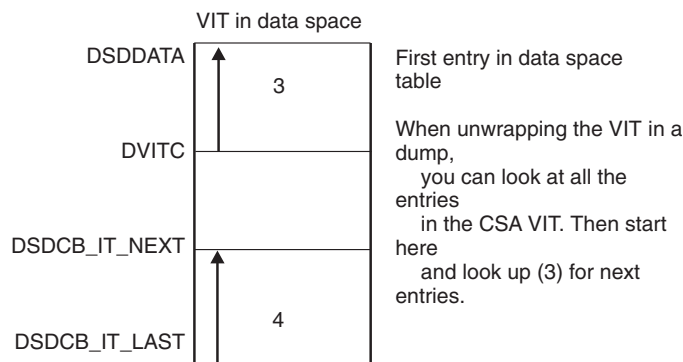
*Figure 5. VIT in data space*

When using the data space option, users must allow for sufficient auxiliary storage. The size of the dump data set will also increase, and users should allow for this increase.

When VTAM recovery routines take a dump, and the data space tracing is active, the data space will be included in the dump. When operators use the MODIFY CSDUMP command to request a dump, the VIT data space will be included in the dump. However, when an operator uses the MVS DUMP command to request a dump, the operator must explicitly specify the VIT data space to have the VIT data space included in the dump.

## Recording traces in external file (MODE=EXT)

If you set MODE=EXT on the MODIFY TRACE command or as a TRACE start option, VTAM allocates (from CSA) the number of 8K VIT buffers specified on the command or start option (default 2). VTAM now accumulates the VIT records in one of these 8K buffers. When a process has insufficient room in the current buffer, it will write the entire 8K buffer to an external trace file using GTF. After the 8K buffer has been written, the process will try again to put its VIT records into the current buffer. If another process finds that there is no current buffer (usually due to another process writing it to GTF), it will try to use one of the preallocated buffers. If there are no other buffers available, either because they are all in the process of being written to GTF or because BFRNUM=0 was specified, VTAM will write the individual records to the external trace file. Avoid using BFRNUM=0 because writing to the external trace file involves large system overhead. Also, allotting an adequate number of 8K buffers prevents individual records from being written to the external trace file. Depending on the situation, VTAM may write the 8K buffers to the external trace file out of order.

The external trace file contains the same information as the internal trace table, except the first-line header record and the BUFF VIT record. The BUFF VIT record is the first VIT record in the 8K buffer. It contains the buffer sequence number, which you can use to determine if the 8K buffers were all written in the proper sequence.

Use external mode when you must collect large amounts of trace data, to lessen the chance of wraparound and lost data. However, unless you are using a tape, the trace data can still fill the external DASD or data file and begin overwriting itself if you do not ensure that the external trace file is large enough.

**Note:** When you specify MODE=EXT, information is still written to the internal trace table for the default options.

The external trace file is produced by GTF, and the default file name is
SYS1.TRACE. You can print the internal trace data with IPCS or TAP. If you use
IPCS to print the data, specify the GTFTRACE option, and set USR(FE1). See *z/OS
Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for
information on using TAP and IPCS to print output and to determine what books
describe TAP and IPCS.

# Module names in internal trace records

Many VTAM internal trace records include the associated module names in
EBCDIC, without the IST prefix and, for some types of trace records, without the
sixth letter. For example, you would see TSSR for module ISTTSCSR. You can save
time by scanning for these module names when you are following the logic flow
through VTAM. You can sometimes isolate a VTAM problem to a specific
component or module without even looking at a dump.

Module names can also be determined from the ISSR field in some VIT records. If
the issuer is an LPA module, the address can be found in the VTAM module list
(which currently contains LPA modules).

The following names are exceptions to the naming convention:

**CPSS**  Session Services for CP-CP Sessions PAB
**DIIO**  Disk I/O PAB
**DSME**
       Directory services management exit PAB
**DSVC**  Directory Services PAB
**LUSS**  Session Services for LU-LU Sessions PAB
**MSTI**  MST Multiple Domain Server Router PAB
**MSTR**  MST Multiple Domain Server Router PAB
**TRS**  Topology Routing Services PAB
**XPRT**  Transaction Program PAB

VTAM component IDs are listed in Table 5.

*Table 5. VTAM component IDs*

| ID | Acronym | Component name |
|---|---|---|
| @@ | | All or multiple components |
| AC | CONFIG | Activation |
| AD | APUNS | Adjacent PU network services |
| AI | API | Application programming interface |
| AM | ADDR-MGT | Address management |
| AP | PSS | Process scheduling services |
| AS | APPC-SVC | APPC services |
| AT | APPNTGMT | APPN transmission group management |
| BS | BFSS | Boundary function session services |
| CD | CDRM | Cross domain resource manager |
| CF | CFS | Coupling facility services |
| CI | OCI | Operator command interface |

# Activating the VIT

*Table 5. VTAM component IDs (continued)*

| ID | Acronym | Component name |
| --- | --- | --- |
| CM | CMIP | Common Management Information Protocol |
| CN | CNS | Common network services |
| CO | COPR | Control operator |
| CP | SSCP | System services control program functions |
| CS | CONFGSVC | Configuration services |
| DE | CONFIG | Deactivation |
| DL | DLUS | Dependent LU server |
| DP | DATACOMP | Data compression |
| DR | DS | Directory services |
| DS | DSS | Data space services |
| EN | ENVIRO | VTAM environment |
| EV | ERVR-MGR | Explicit route virtual route (ER/VR) manager |
| FD | FMT-DMP | Formatted dump |
| FS | CFS | Coupling facility services |
| FU | FUNCTION | SSCP functions |
| GN | SYSGEN | System generation |
| HS | HS | Half-session |
| IE | IEF | Inline exit facility |
| IM | INT-MAP | Internal mappings |
| IN | INIT/TRM | VTAM initialization/ termination |
| IP | SNAIP | IP network access |
| IT | VIT | VTAM internal trace |
| IX | INT-MAC | Internal macroinstructions |
| LL | LLC | Logical link control |
| LM | TPFLOGON | Transaction processing facility (TPF) logon manager |
| LR | DLR | Dump/Load/Restart |
| LS | LSA | Link services architecture |
| LU | LUS | Logical unit services |
| MA | MAINT-SV | Maintenance services |
| MG | MGMTSVC | Maintenance services |
| MS | MGS | Messages |
| MT | MST | Management services transport |
| NA | NAM | Network address management |
| ND | NLDM | Network logical data manager |

*Table 5. VTAM component IDs  (continued)*

| ID | Acronym | Component name |
|---|---|---|
| NO | NOS | Network operator services |
| NR | NRM | Network resource management |
| NS | LNS | LU network services |
| OC | O/C | Open/Close ACB |
| OI | OSI-RPI | VTAM OSI RPI |
| OR | SMS | Storage management services |
| PS | PS | Presentation services |
| PU | PUNS | Physical unit services |
| PV | PVI | Primitive VTAM interface |
| RA | TRACE | Trace services |
| RM | LRM | LU resources manager |
| RV | RVM | MNPS recovery manager |
| SA | SYS-ATTC | System attach |
| SC | SSC | Session services CP-CP |
| SD | SYSDEF | System definition |
| SL | SSL | Session services LU-LU |
| SS | SSCPSS | Session services |
| TA | TACMIP | Topology Agent |
| TB | TSC-BSC | TSC Binary Synchronous Communication |
| TC | TCP-DLC | Data Link Control for TCP |
| TD | TSC-DLC | TSC-DLC |
| TL | TSC-LAN | Token Ring Local Area Network |
| TO | TSO/VTAM | TSO/VTAM |
| TR | TRS | Topology routing services |
| TS | TSC | Transmission subsystem |
| TX | TSX-X25 | X.25 |
| US | USS | Unformatted session services |
| VI | VIO | VSAM I/O |
| VX | VSAMMACS | VSAM user macroinstructions |
| XP | TPS | Transaction program services |
| 62 | APPC | APPC LU6.2 |

## SNAP trace record

Besides recording the previously mentioned functions, you can use the VIT to record 32, 64, 96, or 128 bytes of user-generated information in an SNAP trace. This SNAP trace can place additional information into a VTAM module that might be

useful in resolving a VTAM problem. The VIT traces the specified bytes and enters the data as an SNAP trace entry in the internal trace table or in the external trace file.

Unlike other trace entries, you do not specify the SNAP trace as an option of the MODIFY command. Instead, your code must follow these steps:

1. Verify that the pointer to the ATCVT (ATCLCPTR) is not 0. ATCLCPTR is at low-storage location X'408'.
2. Verify that ATCRACTR is not 0.
3. Call the internal trace recording routine (ISTRACTR) in a VTAM module. Field ATCRACTR in the ATCVT points to the trace recording routine. For the hex offset, see *z/OS Communications Server: SNA Data Areas Volume 1*.

**Note:** SNAP records cannot be taken during the early stages of VTAM initialization, nor can they be taken during the late stages of termination.

The SNAP trace requires the following information in registers 1, 14, and 15 on entry to the trace routine:

**Register 1**
   Points to the 32, 64, 96, or 128 bytes to be recorded.

**Register 14**
   Points to a 2–byte input flag field containing:
   • Hex 141C for a 32-byte SNAP entry
   • Hex 142C for a 64-byte SNAP entry
   • Hex 143C for a 96-byte SNAP entry
   • Hex 144C for a 128-byte SNAP entry

   These flags tell the trace recording routine to record an SNAP trace.

**Two bytes past the address in register 14**
   The address returned to after recording an SNAP trace.

**Register 15**
   Points to the SNAP trace recording routine (ISTRACTR).

When the trace recording routine returns control to the calling module, the contents of registers 6 and 7 are unpredictable. The contents of the other registers, however, are not changed.

The SNAP trace record is unformatted. You should format the first 4 bytes of the SNAP record for a 1- to 4-character EBCDIC record identifier (for example, C"ABCD").

**Notes:**

1. The 1- to 4-character EBCDIC record identifier should not match any existing VIT record identifier.
2. For the VIT snap entry to be used by the VIT analysis tool, the VIT entry name must be the first 2 to 4 characters of the VIT entry.

# Deactivating the VTAM internal trace

*Table 6. Deactivating the VTAM internal trace*

| To deactivate these user selected options: | Specify: |
|---|---|
| Specific options | MODIFY NOTRACE,TYPE=VTAM,OPTION=*options* |
| VIT group options | MODIFY NOTRACE,TYPE=VTAM,OPTION=*group_option*<br><br>All the individual VIT options encompassed by each group option specified by the *group_option* value are deactivated. However, internal tracing continues for the default options that cannot be deactivated; tracing also continues for exception records.<br><br>For example, issuing a MODIFY NOTRACE,TYPE=VTAM,OPTION=EEOPTS command results in the deactivation of these options: CIA, CIO, HPR, MSG, NRM, PIU, PSS, SSCP, SMS, and TCP. But CIO, MSG, NRM, PIU, and SSCP option data continues to be traced internally (PSS and SMS option data is not traced internally). |
| Subtrace options | MODIFY NOTRACE,TYPE=VTAM,SUBTRACE=*subtrace_option*, OPTION=*vit_option* or MODIFY NOTRACE,TYPE=VTAM,SUBTRACE=*subtrace_option*, OPTION=(..,*vit_option*,..)<br><br>The subtrace option specified by the *subtrace_option* value is deactivated. The option specified by the *vit_option* value must relate to the *subtrace_option* value as follows:<br>• If you specify SUBTRACE=ARBP, the *vit_option* value must be HPR or one of the group options that include HPR as an individual option equivalent: DLUROPTS, EEOPTS, HPDTOPTS, HPROPTS, QDIOOPTS, or XCFOPTS.<br>• If you specify SUBTRACE=TGVC or SUBTRACE=TREE, the *vit_option* value must be SSCP or one of the group options (all of which include SSCP as an individual option equivalent): APIOPTS, APPCOPTS, CPCPOPTS, CSMOPTS, DLUROPTS, EEOPTS, HPDTOPTS, HPROPTS, LCSOPTS, QDIOOPTS, STDOPTS, TCPOPTS, or XCFOPTS. |
| Internal only | • MODIFY NOTRACE,TYPE=VTAM,OPTION=END or MODIFY NOTRACE,TYPE=VTAM,OPTION=END,MODE=INT<br>  The internal trace table is reallocated to 999 pages for the default options.<br>• MODIFY NOTRACE,TYPE=VTAM,OPTION=ALL,MODE=INT<br>  The existing internal trace table is used for the default options.<br>• MODIFY NOTRACE,TYPE=VTAM,MODE=INT<br>  External tracing record is stopped. |
| External only | • MODIFY NOTRACE,TYPE=VTAM,OPTION=END,MODE=EXT or MODIFY NOTRACE,TYPE=VTAM,OPTION=ALL,MODE=EXT<br>• MODIFY NOTRACE,TYPE=VTAM,MODE=EXT<br>  External trace recording is stopped and all 8K buffers are freed. |
| Both internal and external | • MODIFY NOTRACE,TYPE=VTAM,OPTION=ALL<br><br>The existing internal trace table is used for the default options. |
| **Notes:**<br>1. Exceptions and default options will continue to be recorded internally.<br>2. If you attempt to stop the VIT using MODIFY NOTRACE, OPTION=END, and the termination hangs, specify MODIFY NOTRACE,OPTION=FORCE to stop the VIT. | |

# Appendix A. VTAM internal trace (VIT) record descriptions

This appendix contains the VTAM internal trace (VIT) record descriptions. The entries are listed alphabetically by entry name.

For more information on VIT options, see "Selecting trace options" on page 10.

## **** Entry for SDUMP taken for CSDUMP request

**Entry:** **** (CSDUMP)
**VIT option:**
>     None (Generated by VTAM)
**Event:** ISTRACSW disabled VIT to issue SDUMPX request for CSDUMP command.
**VIT processing module:**
>     ISTRACTR (SNAP trace recording routine)
**Control is returned to:**
>     ISTRACSW

The **** (CSDUMP) trace record is written when VTAM disables the VTAM internal trace (VIT) to take dump for CSDUMP request.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **** | | | | SDUMP TAKEN FOR CSDUMP REQ | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
>     **Contents**
**00–03**   Record ID: C"****"
**04–1F**   SDUMP TAKEN FOR CSDUMP REQ

## **** Entry for VTAM MODIFY CSDUMP command

**Entry:** **** (CSDUMP)
**VIT option:**
>     None
**Event:** Operator issues MODIFY CSDUMP
**VIT processing module:**
>     ISTRACSW
**Control is returned to:**
>     ISTNOCM9

These trace records are issued when an operator issues the MODIFY CSDUMP command.

**27**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **** | | | | MODIFY CSDUMP COMMAND ISSUED | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
>  **Contents**
**00–03** Record ID: C"****"
**04–1F** MODIFY CSDUMP command issued

# **** Entry for VTAM MODIFY CSDUMP command (continuation)

**Entry:** **** (CSDUMP)
**VIT option:**
>  None
**Event:** Continuation

These records are a continuation of the CSDUMP trace record. These records display the General register content.

**Start record**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **** | | | | MODIFY CSDUMP INVOKERS REGS FOLLOW | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
>  **Contents**
**00–03** Record ID: C"****"
**04–1F** Start delimiter

**First register group record**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **** | | | | R14 | | | | R0 | | | | R1 | | | | R2 | | | | R3 | | | | R4 | | | | R5 | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"****"
**04–07** Contents of register R14
**08–0B** Contents of register R0
**0C–0F** Contents of register R1
**10–13** Contents of register R2
**14–17** Contents of register R3
**18–1B** Contents of register R4
**1C–1F** Contents of register R5

**Second register group record**

| 0 0 0 0 0 1 2 3 | 0 0 0 0 4 5 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|
| **** | R6 | R7 | R8 | R9 | R10 | R11 | R12 |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"****"
**04–07** Contents of register R6
**08–0B** Contents of register R7
**0C–0F** Contents of register R8
**10–13** Contents of register R9
**14–17** Contents of register R10
**18–1B** Contents of register R11
**1C–1F** Contents of register R12

**End record**

| 0 0 0 0 0 1 2 3 | 0 0 0 0 4 5 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|
| **** | END CSDUMP INVOKERS REGS | | | | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"****"
**04–1F** End delimiter

# **** Entry for VTAM ISTORFBA DUMP

> **Entry:** **** (ISTORFBA)

**\*\*\*\* (ISTORFBA)**

**VIT option:**
None (Generated by VTAM)
**Event:** ISTORFBA detected the available buffer count in the BPCB or PXB is incorrect. The registers are included in the VIT and a dump is taken.
**VIT processing module:**
ISTRACTR (SNAP Trace recording routine)
**Control is returned to:**
ISTORFBA

These trace records are issued with SNAP trace entries when ISTORFBA detects the available buffer count in the BPCB or PXB incorrect.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| \*\*\*\* | | | | ISTORFBA DUMP INVOKED | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"\*\*\*\*"
**04–1F** ISTORFBA DUMP invoked

## \*\*\*\* Entry for VTAM ISTORFBA DUMP (continuation)

**Entry:** \*\*\*\* (ISTORFBA)
**VIT option:**
None (Generated by VTAM)
**Event:** Continuation

These records are a continuation of the ISTORFBA DUMP trace record. These records display the General register content.

**Start record**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| \*\*\*\* | | | | ISTORFBA REGS FOLLOW | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"\*\*\*\*"
**04–1F** Start delimiter

**First register group record**

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| **** | R14 | R0 | R1 | R2 | R3 | R4 | R5 |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"****"
**04–07** Contents of register R14
**08–0B** Contents of register R0
**0C–0F** Contents of register R1
**10–13** Contents of register R2 – Save area register 13
**14–17** Contents of register R3
**18–1B** Contents of register R4
**1C–1F** Contents of register R5

**Second register group record**

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| **** | R6 | R7 | R8 | R9 | R10 | R11 | R12 |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"****"
**04–07** Contents of register R6
**08–0B** Contents of register R7
**0C–0F** Contents of register R8
**10–13** Contents of register R9 – BPCB or PXB pointer with wrong available buffer count
**14–17** Contents of register R10
**18–1B** Contents of register R11
**1C–1F** Contents of register R12

**End record**

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| **** | END ISTORFBA REGS | | | | | | |

**\*\*\*\* (ISTORFBA)**

**Byte (hex)**
       **Contents**
**00–03**   Record ID: C"\*\*\*\*"
**04–1F**   End delimiter

## \*\*\*\* Entry for VTAM FFST

**Entry:** **\*\*\*\* (FFST)**
**VIT option:**
       None (Generated by VTAM)
**Event:** VTAM probe triggered
**VIT processing module:**
       ISTRACTR (SNAP trace recording routine)
**Control is returned to:**
       ISTRACZE

The \*\*\*\* (FFST) trace record is written when VTAM disables the VTAM internal
trace (VIT) to collect information for a probe that has been triggered. This entry is
not associated with any VIT options but is recorded when an unusual condition
triggers a probe instruction.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 0 0 0 | 0 | 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 | 1 |
| 0 1 2 3 | 4 | 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E | F |
| \*\*\*\* |  | 'VIT DISABLED FOR VTAM/FFST' |  |

**Byte (hex)**
       **Contents**
**00–03**   Record ID: C"\*\*\*\*"
**04**       Blank
**05–1E**   C"VIT DISABLED FOR VTAM/FFST"
**1F**       Blank

## \*\*\*\* Entry for VTAM Phantom FFST

**Entry:** **\*\*\*\* (Phantom FFST)**
**VIT option:**
       None
**Event:** FFST IPROBE trip in programming product
**VIT processing module:**
       • VTAM — ISTRAZPF

       • CSM — IVTSMZPF
**Control is returned to:**
       Module that issued the IPROBE macro.

These trace records are issued when an IPROBE is tripped, and First Failure
Support Technology (FFST) is not active.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **** | | | | PFFST INVOKED FOR probe name | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C″****″

**04–1F** Probe name in first record identifies what product. For example:

- ISTxxxxx is for VTAM
- IVTxxxxx is for CSM

# **** Entry for VTAM Phantom FFST (continuation)

**Entry:** **** (Phantom FFST)

**VIT option:**
> None

**Event:** Continuation

These records are a continuation of the PFFST trace record. These records display the General register content of the module that attempted the IPROBE (general registers R14, R0 through R12, followed by a delimiter record).

**Start record**

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **** | | | | PFFST INVOKER REGS FOLLOW | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C″****″

**04–1F** Start delimiter

**First register group record**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **** | | | | R14 | | | | R0 | | | | R1 | | | | R2 | | | | R3 | | | | R4 | | | | R5 | | | |

**Byte (hex)**
  **Contents**
**00–03**  Record ID: C"****"
**04–07**  Contents of register R14
**08–0B**  Contents of register R0
**0C–0F**  Contents of register R1
**10–13**  Contents of register R2
**14–17**  Contents of register R3
**18–1B**  Contents of register R4
**1C–1F**  Contents of register R5

**Second register group record**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **** | | | | R6 | | | | R7 | | | | R8 | | | | R9 | | | | R10 | | | | R11 | | | | R12 | | | |

**Byte (hex)**
  **Contents**
**00–03**  Record ID: C"****"
**04–07**  Contents of register R6
**08–0B**  Contents of register R7
**0C–0F**  Contents of register R8
**10–13**  Contents of register R9
**14–17**  Contents of register R10
**18–1B**  Contents of register R11
**1C–1F**  Contents of register R12

**End record**

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| **** | END PFFST INVOKERS REGS | |

**Byte (hex)**

    **Contents**

**00–03**  Record ID: C"****"

**04–1F**  End delimiter

ATCPFSAV, located in the ATCVT control block, contains the callers Register 13. This pointer field can be located in the PFFST dump.

## ABND entry for abend SNAP routine

**Entry:  ABND**

**VIT option:**

    None (Generated by SNAP routine)

**Event:  Abend**

**VIT processing module:**

    ISTRACTR (SNAP trace recording routine)

**Control is returned to:**

    VTAM abend recovery routine (many possible)

This trace record is written when an abend occurs in a VTAM module. This entry is not associated with any VIT options but is recorded as an exception condition when an abend occurs.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F |
| ABND | ABEND CODE | PST ADDRESS OR 0 | RPH ADDRESS OR 0 | SDWA ADDRESS OR 0 | SDWA FLAGS OR 0 | ABEND PSW OR 0 |

**Byte (hex)**

    **Contents**

**00–03**  Record ID: C"ABND"

**04–07**  Abend completion code

**08–0B**  PST address or 0

**0C–0F**  Request parameter header (RPH) address or 0

**10–13**  SDWA address or 0

**14–17**  SDWA flags or 0

**18–1F**  Abend PSW or 0

## ACA1 or ACI1 entry for LU 6.2 authorized IO or LU 6.2 TPIO (Part 1)

**Entry:** ACA1 or ACI1
**VIT option:**
   APPC
**Event:** LU 6.2 authorized IO or LU 6.2 TPIO (Part 1)
**VIT processing module:**
   ISTRACAC
**Control is returned to:**
   ISTAICAR

The ACA1 trace record shows LU 6.2 requests under the authorized path or LU 6.2 requests not under the authorized path but issued by programs running under an authorized key. The ACI1 trace record shows LU 6.2 requests running under a normal path.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C | 0D | 0E | 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACA1 OR ACI1 | | | | ID | E X T D S | O P T 1 | O P T 6 | RPL ADDRESS | C N T R L | Q U A L | B U F L | | RPL AREA | APPCCMD FLAGS OR 0 | USR FLD, CID, CONVID, OR 0 | SENSE DATA OR 0 |

**Byte (hex)**
   **Contents**
**00–03**   Record ID:
   - C"ACA1" for authorized path
   - C"ACI1" for normal path
**04**   ID is the primary address space ID (ASID). This field is 0 if the ID is greater than X'FF'.
**05**   Exit definition (RPLEXTDS)
**06**   Option code byte 1 (RPLOPT1)
**07**   Option code byte 6 (RPLOPT6)

   Contains the RPLXBFL flag (bit 5) that indicates whether the application is requesting HPDT services.
**08–0B**   RPL address
**0C**   APPCCMD CONTROL operand value:
   **X'10'**   ALLOC
   **X'11'**   PREALLOC
   **X'12'**   SENDFMH5
   **X'20'**   RESETRCV
   **X'30'**   DEALLOC
   **X'31'**   DEALLOCQ
   **X'40'**   OPRCNTL
   **X'50'**   PREPRCV
   **X'60'**   RCVFMH5
   **X'70'**   RECEIVE
   **X'71'**   RCVEXPD
   **X'80'**   REJECT
   **X'90'**   SEND
   **X'91'**   SENDEXPD
   **X'92'**   SENDRCV
   **X'A0'**   SETSESS

|       |       |          |
|-------|-------|----------|
|       | **X'B0'** | TESTSTAT |
| **0D** |       | APPCCMD QUALIFY operand value: |
|       | **X'00'** | NULL |
|       | **X'01'** | ABNDPROG |
|       | **X'02'** | ABNDSERV |
|       | **X'03'** | ABNDTIME |
|       | **X'04'** | ABNDUSER |
|       | **X'05'** | ANY |
|       | **X'06'** | CNOS |
|       | **X'07'** | CONFIRM |
|       | **X'08'** | CONFRMD |
|       | **X'09'** | DATA |
|       | **X'0A'** | DATACON |
|       | **X'0B'** | DATAFLU |
|       | **X'0C'** | DEFINE |
|       | **X'0D'** | DISPLAY |
|       | **X'0E'** | ERROR |
|       | **X'0F'** | FLUSH |
|       | **X'10'** | RQSEND |
|       | **X'11'** | SPEC |
|       | **X'12'** | ACTSESS |
|       | **X'13'** | DACTSESS |
|       | **X'14'** | ALLOCD |
|       | **X'15'** | IMMED |
|       | **X'16'** | CONWIN |
|       | **X'17'** | SESSION |
|       | **X'18'** | CONV |
|       | **X'19'** | SUSPEND |
|       | **X'1A'** | RESUME |
|       | **X'1B'** | RESTORE |
|       | **X'1C'** | SYNCBEG |
|       | **X'1D'** | SYNCEND |
|       | **X'1E'** | CONVGRP |
|       | **X'1F'** | WHENFREE |
|       | **X'20'** | IANY |
|       | **X'21'** | ISPEC |
|       | **X'22'** | ALL |
|       | **X'23'** | IALL |
|       | **X'24'** | QUEUE |
|       | **X'25'** | DATAQUE |

**0E–0F** User buffer length (RPLBUFL) for receive RPL or 0

**10–13** Address of SEND data or RECEIVE buffer

If HPDT services are requested, the RPL area contains the extended buffer list area. The contents are recorded in the XBA1, XBA2, and XBA3 entries.

**14–17** APPCCMD flags (RPL6FLGS) or 0

**18–1B** User field for LU 6.2 commands when CONTROL equals ALLOC, PREALLOC, or RCVFMH5. CID for LU 6.2 commands when CONTROL equals OPRCNTL, and QUALIFY equals either ACTSESS or DACTSESS. 0 for LU 6.2 commands for which one of the following is true:
- CONTROL equals OPRCNTL, and QUALIFY equals either CNOS, DEFINE, or DISPLAY.
- CONTROL equals RECEIVE or RCVDEXPD, and QUALIFY equals ANY or IANY.
- CONTROL equals REJECT, and QUALIFY equals SESSION or CONVGRP.

- CONTROL equals SETSESS, and QUALIFY equals RESUME.

Conversation ID for all other LU 6.2 commands

**1C–1F** Sense data (RPL6SNSO) or 0

## ACA2 or ACI2 entry for LU 6.2 authorized IO or LU 6.2 TPIO (Part 2)

**Entry: ACA2 or ACI2**
**VIT option:**
APPC
**Event:** LU 6.2 authorized IO or LU 6.2 TPIO (Part 2)
**VIT processing module:**
ISTRACAC

The ACA2 trace record is a continuation of ACA1. The ACI2 trace record is a continuation of ACI1.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| ACA2 OR ACI2 | RETURN ADDRESS | SEE DESCRIPTION | LOCAL LU NAME, 0, OR BLANKS | PARTNER LU NAME, 0, OR BLANKS |

**Byte (hex)**
**Contents**
**00–03** Record ID:
- C"ACA2" for continuation of ACA1
- C"ACI2" for continuation of ACI1

**04–07** Address of the issuer of the APPCCMD macroinstruction
**08–0F**
- When CONTROL=ALLOC, PREALLOC, or REJECT, and QUALIFY=CONVGRP, then 08–0B is the conversion group ID (CGID), and 0C–0F is 0.
- When CONTROL equals ALLOC or PREALLOC (except when QUALIFY equals CONVGRP) or when CONTROL equals OPRCNTL, and QUALIFY equals either CNOS, DEFINE, or DISPLAY, then 08–0F is the mode name for LU 6.2 commands.
- When CONTROL equals SETSESS or when CONTROL equals REJECT, and QUALIFY equals SESSION, then 08–0F is the session instance identifier for LU 6.2 commands.

**10–17** Local logical unit name for LU 6.2 commands when CONTROL equals ALLOC or PREALLOC or when CONTROL equals OPRCNTL, and QUALIFY equals either CNOS, DEFINE, DISPLAY, or RESTORE. Otherwise, this field contains blanks or 0.
**18–1F** Partner logical unit name for LU 6.2 commands when CONTROL equals ALLOC or PREALLOC or when CONTROL equals OPRCNTL, and QUALIFY equals either CNOS, DEFINE, DISPLAY or RESTORE. Otherwise, this field contains blanks or 0.

## ACA3 or ACI3 entry for LU 6.2 authorized IO or LU 6.2 TPIO (Part 3)

**Entry: ACA3 or ACI3**

**VIT option:**
 APPC
**Event:** LU 6.2 authorized IO or LU 6.2 TPIO (Part 3)
**VIT processing module:**
 ISTRACAC

The ACA3 trace record is a continuation of ACA2. The ACI3 trace record is a continuation of ACI2.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| ACA3 OR ACI3 | | | | 0 | | | | | | | | | | | | | | | | | | | | PARTNER NETID, 0, OR BLANKS | | | | | | | |

**Byte (hex)**
 **Contents**
**00–03** Record ID:
  • C"ACA3" for continuation of ACA2
  • C"ACI3" for continuation of ACI2
**04–17** 0
**18–1F** Partner network identifier for the LU 6.2 commands when CONTROL equals ALLOC or PREALLOC (except for QUALIFY=CONVGRP) or when CONTROL equals OPRCNTL, and QUALIFY equals either CNOS, DEFINE, DISPLAY, or RESTORE. Otherwise, this field contains blanks or 0.

# ACP1 or ACR1 entry for LU 6.2 user post or RPL exit (Part 1)

**Entry:** ACP1 or ACR1
**VIT option:**
 APPC
**Event:** LU 6.2 user post or RPL exit (Part 1)
**VIT processing module:**
 ISTRACAC
**Control is returned to:**
 ISTAICPT and ISTAPCSX for ACP1

 ISTAPCSX and ISTAICRX for ACR1

The ACP1 trace record is written when an ECB is posted. The ACR1 trace record is written when an RPL exit is dispatched. These records signal that the APPCCMD macro has completed execution and show the data returned to the user application program.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| ACP1 OR ACR1 | | | | ID | RTUN | RTNCD | FDB2 | RPL ADDRESS | | | | CNTRL | QUAL | OPT6 | 0 | RCPRI | | RCSEC | | APPCCMD FLAGS | | | | CONV ID OR 0 | | | | SENSE DATA OR 0 | | | |

**Byte (hex)**

**Contents**

00–03 Record ID:
  - C"ACP1" for ECB posting
  - C"ACR1" for RPL exit dispatching

04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05 APPCCMD returned indicators (RPL6RTUN)

**Bit** **Meaning**

**1... ....**
  FMH5 received indicator

**.1.. ....**
  LOGON received indicator

**..1. ....**
  SIGNAL received indicator

06 VTAM return code (RPLRTNCD)

07 VTAM feedback code (RPLFDB2)

08–0B RPL address

0C APPCCMD CONTROL operand value:

**X'10'** ALLOC
**X'11'** PREALLOC
**X'12'** SENDFMH5
**X'20'** RESETRCV
**X'30'** DEALLOC
**X'31'** DEALLOCQ
**X'40'** OPRCNTL
**X'50'** PREPRCV
**X'60'** RCVFMH5
**X'70'** RECEIVE
**X'71'** RCVEXPD
**X'80'** REJECT
**X'90'** SEND
**X'91'** SENDEXPD
**X'92'** SENDRCV
**X'A0'** SETSESS
**X'B0'** TESTSTAT

0D APPCCMD QUALIFY operand value:

**X'00'** NULL
**X'01'** ABNDPROG
**X'02'** ABNDSERV
**X'03'** ABNDTIME
**X'04'** ABNDUSER
**X'05'** ANY
**X'06'** CNOS
**X'07'** CONFIRM
**X'08'** CONFRMD
**X'09'** DATA
**X'0A'** DATACON
**X'0B'** DATAFLU
**X'0C'** DEFINE
**X'0D'** DISPLAY
**X'0E'** ERROR
**X'0F'** FLUSH
**X'10'** RQSEND
**X'11'** SPEC

|        |          |
|--------|----------|
| **X'12'** | ACTSESS  |
| **X'13'** | DACTSESS |
| **X'14'** | ALLOCD   |
| **X'15'** | IMMED    |
| **X'16'** | CONWIN   |
| **X'17'** | SESSION  |
| **X'18'** | CONV     |
| **X'19'** | SUSPEND  |
| **X'1A'** | RESUME   |
| **X'1B'** | RESTORE  |
| **X'1C'** | SYNCBEG  |
| **X'1D'** | SYNCEND  |
| **X'1E'** | CONVGRP  |
| **X'1F'** | WHENFREE |
| **X'20'** | IANY     |
| **X'21'** | ISPEC    |
| **X'22'** | ALL      |
| **X'23'** | IALL     |
| **X'24'** | QUEUE    |
| **X'25'** | DATAQUE  |

**0E**   Option code byte 6 (RPLOPT6)

Contains the RPLXBFL flag (bit 5), which indicates whether the application is requesting HPDT services.

**0F**   0

**10–11**  LU 6.2 primary return code, RCPRI (RPL6RCPR field in the RPL extension). For more information about RCPRI return codes, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Reference*.

**12–13**  LU 6.2 secondary return code, RCSEC (RPL6RCSC field in the RPL extension). For more information about RCSEC return codes, refer to *z/OS Communications Server: SNA Programmer's LU 6.2 Reference*.

**14–17**  APPCCMD flags (RPL6FLGS)

**18–1B**  Conversation ID or 0

**1C–1F**  Sense data returned (RPL6SNSI) or 0

# ACP2 or ACR2 entry for LU 6.2 user post or RPL exit (Part 2)

**Entry:  ACP2 or ACR2**
**VIT option:**
   APPC
**Event:  LU 6.2 user post or RPL exit (Part 2)**
**VIT processing module:**
   ISTRACAC

The ACP2 trace record is a continuation of the ACP1 trace record. The ACR2 trace record is a continuation of the ACR1 trace record.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| ACP2 OR ACR2 | | ECB ADDRESS OR EXIT ADDRESS | | RPL6 ADDRESS | | C N T R L | R P L E X T D S | RCV IN OR 0 | | RPL AREA | | RPL RECORD LENGTH | | CGID | | C N V F S M | R T R F S M | E R R F S M | S R T R F S |

**Byte (hex)**

**Contents**

**00–03** Record ID:
- C"ACP2" for continuation of ACP1
- C"ACR2" for continuation of ACR1

**04–07** ECB address for ACP2. RPL exit address for ACR2

**08–0B** RPL6 address

**0C** APPCCMD CONTROL operand value:

| Bit | Meaning |
|---|---|
| X'10' | ALLOC |
| X'11' | PREALLOC |
| X'12' | SENDFMH5 |
| X'20' | RESETRCV |
| X'30' | DEALLOC |
| X'31' | DEALLOCQ |
| X'40' | OPRCNTL |
| X'50' | PREPRCV |
| X'60' | RCVFMH5 |
| X'70' | RECEIVE |
| X'71' | RCVEXPD |
| X'80' | REJECT |
| X'90' | SEND |
| X'91' | SENDEXPD |
| X'92' | SENDRCV |
| X'A0' | SETSESS |
| X'B0' | TESTSTAT |

**0D** RPLEXTDS contains the RPLXSRV flag (bit 7), which is set if VTAM accepts all CSM buffers from the application. If no CSM buffers are specified by the application, this field is not set.

**0E–0F** Indicators for the type of information received or 0

| Bit | Meaning |
|---|---|
| 1... .... | DATA for byte 0E; PARTIAL_PS_HEADER for byte 0F |
| .1.. .... | DATA_COMPLETE |
| ..1. .... | DATA_INCOMPLETE |
| ...1 .... | SEND |
| .... 1... | CONFIRM |
| .... .1.. | DEALLOCATE |

```
.... ..1.
     LOG_DATA
.... ...1
     PS_HEADER
1... ....
     PARTIAL_PS_HEADER
.xxx xxxx
     0
```

**10–13**   Address of SEND data or RECEIVE buffer (RPLAREA)

If OPTCD=XBUFLST is specified, this field contains the extended buffer list area. The contents of the extended buffer list are recorded in the XBA1, XBA2, XBA3 entries.

**14–17**   RPL record length

**18–1B**   Conversation group ID (CGID) when CONTROL equals ALLOC, PREALLOC, or RCVFMH5; otherwise, 0

**1C**   RAB conversation FSM

**1D**   RAB router FSM

**1E**   RAB error or failure FSM

**1F**   RAB saved router FSM

# ACRC entry for RPL6 return code

**Entry:**  **ACRC**
**VIT option:**
    APPC
**Event:**  APPCSNRC macro
**VIT processing module:**
    ISTRACAC
**Control is returned to:**
    Issuer of the APPCSNRC macro

This trace record is written when an APPC module issues a macroinstruction to set an RPL6RC nonzero return code. The condition that this entry records is an exception and is recorded regardless of whether or not the APPC VIT option is active.

| 00 01 02 03 | 04 | 05 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| ACRC | ID | 0 | INSTANCE ID | CONVERSATION ID | HALF-SESSION ID | MODULE NAME | CONTROL BLOCK ADDRESS | RETURN CODE | RPH ADDRESS |

**Byte (hex)**
    **Contents**

**00–03**   Record ID: C"ACRC"

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–06** 0
**07** Instance identifier in invoking module
**08–0B** Conversation ID
**0C–0F** Half-session ID
**10–13** Invoking module name
**14–17** Address of control block containing return code
**18–1B** Return code (RPL6RC)
    **18–19** Primary return code (RPL6RCPR)
    **1A–1B** Secondary return code (RPL6RCSC)
**1C–1F** Request parameter header (RPH) address

# ACSN entry for APPC sense code

**Entry:  ACSN**
**VIT option:**
    APPC
**Event:  APPCSNRC macro**
**VIT processing module:**
    ISTRACAC
**Control is returned to:**
    Issuer of the APPCSNRC macro

This trace record is written when an APPC module issues a macroinstruction to set a nonzero sense code. It is treated as an exception condition and is always traced if the VIT is active, regardless of the VIT options specified.

| 00 01 02 03 | 04 | 05 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| ACSN | ID | 0 | INSTANCE ID | CONVERSATION ID | HALF-SESSION ID | MODULE NAME | CONTROL BLOCK ADDRESS | SENSE CODE OR RETURN CODE | RPH ADDRESS |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"ACSN"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–06** 0
**07** Instance identifier in invoking module
**08–0B** Conversation ID
**0C–0F** Half-session ID
**10–13** Invoking module name
**14–17** Address of control block containing sense code
**18–1B** Sense code or internal return code
**1C–1F** Request parameter header (RPH) address

## ACU1 entry for LU 6.2 user exit (Part 1)

**Entry:** ACU1
**VIT option:**
APPC
**Event:** LU 6.2 user exit (Part 1)
**VIT processing module:**
ISTRACAC
**Control is returned to:**
ISTAICUE

This trace record is written when an LU 6.2 user exit (either ATTN or TPEND) is dispatched. If both the API and APPC trace options are active, and a TPEND user exit is dispatched, VTAM generates user exit trace records for both API and APPC.

| 0 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| ACU1 | I D | C O D E | 0 | EXIT TYPE DEPEND INFO | 0 | PLU NAME OR APPLICATION ID | SLU NAME OR 0 |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"ACU1"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Exit code for user exit:
X'0C' for ATTN
X'0D' for TPEND

**06–07** 0

**08–0B** The type of exit determines the value of this field.

- For ATTN exit, the exit subtype (CNOS, FMH5, or LOSS)

- For TPEND exit, byte 8 is the reason code:

  **Code** **Description**

  **X'00'** Standard HALT command issued to close the network.

  **X'04'** HALT QUICK command or VARY INACT,I or F command issued for the application program name.

  **X'08'** HALT CANCEL command issued or VTAM terminated abnormally.

  **X'0C'** Alternate application issued an OPEN ACB for the same ACB this application has opened.

**0C–0F** 0

**10–17** PLU name for ATTN exit; application program ID for TPEND exit

**18–1F** SLU name for ATTN exit; 0 for TPEND exit

## ACU2 entry for LU 6.2 user exit (Part 2)

**Entry:** ACU2
**VIT option:**
APPC
**Event:** LU 6.2 user exit (Part 2)

**VIT processing module:**
        ISTRACAC

This trace record is a continuation of the ACU1 trace record.

| 0 0 0 0<br>0 1 2 3 | 0 0 0 0<br>4 5 6 7 | 0 0 0 0 0 0 0 0<br>8 9 A B C D E F | 1 1 1 1 1 1 1 1<br>0 1 2 3 4 5 6 7 | 1 1 1 1 1 1 1 1<br>8 9 A B C D E F |
|---|---|---|---|---|
| ACU2 | EXIT ADDRESS | MODE NAME OR 0 | 0 | NETID OR 0 |

**Byte (hex)**
        **Contents**
**00–03**  Record ID: C"ACU2"
**04–07**  Exit address
**08–0F**  Mode name for ATTN exit; 0 for TPEND exit
**10–17**  0
**18–1F**  Network identifier for ATTN exit; 0 for TPEND exit

# ADE entry for device errors

**Entry:  ADE**
**VIT option:**
        CIO
**Event:**  Failure occurs when trying to activate a device (Instance 0), deactivate a device (Instance 1), allocate a device (Instance 2), or deallocate a device (Instance 2).
**VIT processing module:**
        ISTRACCI
**Control is returned to:**
        ISTINCAV

This trace record contains information about an error when VTAM attempts to activate a device (Instance 0), deactivate a device (Instance 1), allocate a device (Instance 2), or deallocate a device (Instance 2). The three instances of the trace record follow.

This record is treated as an exception entry and is always traced regardless of the VIT options specified.

**Instance 0**

| 0 0 0 0<br>0 1 2 3 | 0 0 0 0<br>4 5 6 7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1 1 1<br>0 1 2 3 | 1 1 1 1<br>4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|
| ADE | C O D E | A S I D | 0 | R C | T Y P E | DLR CUA | U C B R C | R E A S O N | DEV ADD | F L A G S | XCNCB ADDRESS | SRB ADDRESS | IOSB ADDRESS | RPH ADDRESS |

**Byte (hex)**        **Contents**
**00–02**                Record ID: C"ADE"

| Byte (hex) | Contents |
|---|---|
| **03** | Code: X'F0' for activate device (DLRPLFCD) |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Return code (DLRPLRCD) |
| **07** | Device type |
| **08–09** | Channel unit address |
| **0A** | UCBINFO return code or 0 |
| **0B** | UCBINFO reason code or 0 |
| **0C–0D** | Data device address (DLRDATAD) |
| **0E–0F** | Flags: |

| Bit | Meaning |
|---|---|
| 1... .... .... .... | MPC indicator |
| .1.. .... .... .... | TCP/IP legacy DLC |
| ..1. .... .... .... | IDX indicator |
| ...1 .... .... .... | TCP/IP CTC DLC |
| .... 1... .... .... | TCP/IP CLAW DLC |
| .... .1.. .... .... | TCP/IP LCS DLC |
| .... ..1. .... .... | TCP/IP CDLC DLC |
| .... ...1 .... .... | TCP/IP HYPERchannel DLC |
| .... .... 1... .... | QDIO indicator |
| .... .... .111 1111 | Not used |

| Byte (hex) | Contents |
|---|---|
| **10–13** | XCNCB address |
| **14–17** | SRB address |
| **18–1B** | IOSB address |
| **1C–1F** | RPH address |

**Instance 1**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADE | | | CODE | ASID | 0 | RC | TYPE | DLR CUA | | 0 | | | | FLAGS | ATNIX | DCB ADDRESS | | | | 0 | | | | | | | | RPH ADDRESS | | | |

| Byte (hex) | Contents |
|---|---|
| **00–02** | Record ID: C"ADE" |
| **03** | Code: X'F1' for deactivate device (DLRPLFCD) |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Return code (DLRPLRCD) |
| **07** | Device type |
| **08–09** | Channel unit address |
| **0A–0D** | 0 |
| **0E** | Flags: |

| Bit | Meaning |
|---|---|
| 1... .... | MPC indicator |
| .1.. .... | TCP/IP legacy DLC |
| ..1. .... | IDX indicator |

| | | |
|---|---|---|
| **...1 1111** | Not used | |
| **0F** | UCB attention table index | |
| **10–13** | DCB address | |
| **14–1B** | 0 | |
| **1C–1F** | RPH address | |

**Instance 2**

| 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 0 | 0 | 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|-------|---|---|---|---|---|-----|---|---|---------|---------|---------|---------|---------|
| 0 1 2 | 3 | 4 | 5 | 6 | 7 | 8 9 | A | B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| ADE | CODE | ASID | 0 | RC | TYPE | DEV ADD | ENTRY | HEADER | LINE RDTE ADDRESS | UCB ADDRESS | RUPE ADDRESS | RUPE RU CODE OR 0 | RPH ADDRESS |

| Byte (hex) | Contents |
|---|---|
| **00–02** | Record ID: C"ADE" |
| **03** | Code: X'CA' for allocate device or X'CD' for deallocate device (DLRPLFCD) |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Return code (DLRPLRCD) |
| **07** | Device type |
| **08–09** | Device address |
| **0A** | RDTE entry type or 0 |
| **0B** | RDTE header type or 0 |
| **0C–0F** | Line RDTE address |
| **10–13** | UCB address |
| **14–17** | RUPE address |
| **18–1B** | RUPE RU code or 0 |
| **1C–1F** | RPH address |

# AFSM entry for altering an FSM state

**Entry: AFSM**
**VIT option:**
    SSCP
**Event:** Alteration of an FSM state
**VIT processing module:**
    ISTRACSC
**Control is returned to:**
    The module that issued the INTRACE macroinstruction

This trace record is written when the current state of a FSM changes.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 | 0 | 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---------|---|---|-----|---|---|---|---------|---------|---------|---------|---------|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 | 9 | A | B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| AFSM | A S I D | 0 | D L C T Y P E | F S M T Y P E | O L D S T A T E | N E W S T A T E | W K E T Y P E / I N S T A N C E / R E S O U R C E A D D R E S S | RETURN ADDRESS | WORK ELEMENT ADDRESS | MODULE NAME | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"AFSM"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Data link control type:

   **A** ATM

   **E** Enterprise Extender

   **L** LAN (External Communication Adapter)

**07** FSM type:

   **A** AAL FSM (ATM only)

   **D** LDLC FSM

   **E** LDLC XID FSM

   **L** Link FSM

   **P** Port FSM

   **X** XID FSM

**08** Old state

**09** New state

**0A** Work element type:

   **01** ISTRPH

   **17** ISTAUCPL

   **40** ISTLSPL

   **54** ISTRUPE

   **58** ISTTQE

   **99** IUTTIPAC

   **9A** ISTTSPL

   **9B** ISTLSCB

**0B** 0 or instance of the SETAFSM macro in the module

**0C–0F** Address of the control block containing the FSM

**10–13** Return address of the module that changed the FSM state

**14–17** Work element address

**18–1B** Name of the module that changed the FSM state

**1C–1F** Request parameter header (RPH) address

---

# AI1 entry for authorized IO (Part 1)

**Entry: AI1**

**VIT option:**

   API

**Event:** Authorized IO (Part 1)

**VIT processing module:**

   ISTRACAP

**Control is returned to:**

> **ISTAICIR for:**
> > **ISTOCCSM for:**
> **SEND** INQUIRE
> **RECEIVE**
> > INTRPRET
> **RESETSR**
> > RCVCMD
> **SESSIONC**
> > SENDCMD
> > SETLOGON
> > SESSIONC (BIND)
> > SIMLOGON
> > OPNDST
> > REQSESS
> > OPNSEC
> > TERMSESS
> > CLSDST

This trace record shows API requests under the authorized path or API requests not under the authorized path but issued by programs running under an authorized key. The API routine ISTAICIR or ISTOCCSM writes the entry and then queues the RPL to the correct PAB. For an explanation of the RPL fields, refer to *z/OS Communications Server: SNA Data Areas Volume 1*.

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 | 1 | 1 | 1 | 1 | 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 0 1 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 9 A B | 0 C D E F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 6 7 | 1 8 9 A B | 1 C D E F |
| AI1 | R E Q T Y | I D | E X T D S | O P T 4 | O P T 1 | RPL ADDRESS | NIB ADDR OR CID | R H 3 | S R T Y P | V T F L 1 | V T F L 2 | C H N | RU CNTL CODE | OPTC2 | OPTC3 |

**Byte (hex)**
> **Contents**

**00–02**   Record ID: C"AI1"

**03**   RPL request type (RPLREQ)
> **X'15'**   SETLOGON
> **X'16'**   SIMLOGON
> **X'17'**   OPNDST
> **X'19'**   CHANGE
> **X'1A'**   INQUIRE
> **X'1B'**   INTRPRET
> **X'1F'**   CLSDST
> **X'22'**   SEND
> **X'23'**   RECEIVE
> **X'24'**   RESETSR
> **X'25'**   SESSIONC
> **X'27'**   SENDCMD
> **X'28'**   RCVCMD
> **X'29'**   REQSESS
> **X'2A'**   OPNSEC
> **X'2C'**   TERMSESS

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

| | | |
|---|---|---|
| **05** | Exit definition (RPLEXTDS) | |
| **06** | Option code byte 4 (RPLOPT4) | |
| **07** | Option code byte 1 (RPLOPT1) | |
| **08–0B** | RPL address | |
| **0C–0F** | NIB address or CID | |
| **10** | Third byte of RH (RPLRH3) | |
| **11** | Send/receive type (RPLSRTYP) | |
| **12** | VTAM flags (RPLVTFL1) | |
| **13** | Post/respond flags (RPLVTFL2) | |
| **14** | RU chain position (RPLCHN) | |
| **15–17** | RU control codes (RPLCNTRL) | |
| **18–1B** | VTAM options (RPLOPTC2, which corresponds to RPLOPT5 through RPLOPT8) (Refer to *z/OS Communications Server: SNA Programming* for additional information.) | |
| **1C–1F** | VTAM options (RPLOPTC3, which corresponds to RPLOPT9 through RPLOPT12) (Refer to *z/OS Communications Server: SNA Programming* for additional information.) | |

## AI2 entry for authorized IO (Part 2)

**Entry: AI2**
**VIT option:**
> API

**Event:** Authorized IO (Part 2)
**VIT processing module:**
> ISTRACAP

This trace record is a continuation of the AI1 entry. It shows additional information about the PLU and SLU in a session established or terminated by a SIMLOGON, OPNDST, CLSDST, REQSESS, OPNSEC, or TERMSESS macroinstruction. This information includes some of the parameters exchanged and the names of the PLU and SLU.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| AI2 0 | RETURN ADDRESS | LOGON MODE NAME OR 0 | APPLICATION LU NAME (IF AVAILABLE) | PARTNER LU (IF AVAILABLE) |

**Byte (hex)**
> **Contents**

| | |
|---|---|
| **00–02** | Record ID: C"AI2" |
| **03** | 0 |
| **04–07** | Address of the issuer of the macro |
| **08–0F** | For SIMLOGON, OPNDST, CLSDST, or REQSESS: Logon mode name (NIBLMODE) or 0 if the NIB is not available |
| | For OPNSEC or TERMSESS: 0 |
| **10–17** | Application LU name if available |
| **18–1F** | Partner LU name if available |

## AI3 entry for authorized IO (part 3)

**Entry: AI3**

**VIT option:**
　　　API
**Event:** Authorized IO (Part 3)
**VIT processing module:**
　　　ISTRACAP

This trace record is a continuation of the AI2 entry. It shows additional information in a session established or terminated by a SIMLOGON, OPNDST, CLSDST, REQSESS, or OPNSEC macroinstruction.

| 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| AI3 | 0 | ADDRESS OF USER DATA, BIND, OR 0 | TARGET LU NETID OR<br><br>N C　N R C<br>I O　I P O<br>B U　B A U<br>　N　　R N<br>　T　　M T　　0 | TARGET LU NAME (IF AVAILABLE) OR 0 | PARTNER LU NETID (IF AVAILABLE) |

**Byte (hex)**
　　　**Contents**
**00–02** Record ID: C"AI3"
**03** 　 0
**04–07** For SIMLOGON, CLSDST, or REQSESS: Address of user data (RPLAREA)

　　　For OPNDST or OPNSEC: Address of BIND (NIBNDAR) or 0 if the NIB is not available
**08–0F** Target LU network identifier (when for CLSDST PASS) or:

　　　**Byte (hex)**
　　　　　**Contents**
　　　**08–09** Count of node initialization blocks (NIBs) in NIB list (when not for CLSDST PASS)
　　　**0A–0B** Count of node initialization blocks (NIBs) in NIB list with NIBRPARM=0 (when not for CLSDST PASS)
　　　**0C–0F** 　0
**10–17** Target logical unit name, if available, or 0
**18–1F** Partner LU network identifier, if available

# ALSx entry for adjacent link station

**Entry:** **ALSA, ALSD, ALSP, ALSR, ALSS, or ALSV**
**VIT option:**
　　　SSCP
**Event:** ALSLIST macro
**VIT processing module:**
　　　ISTRACSC
**Control is returned to:**
　　　Module invoking the INTRACE macro that caused the record to be produced.

The adjacent link station (ALS) trace record contains information about the adding, deleting, replacing, selecting, or validating of an adjacent link station.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F |
| ALSA ALSD ALSP ALSR ALSS ALSV | I D | R C | C B I D | F L A G S | NEWALS OR SELALS OR 0 | RETURN ADDR | CDRADDR | OLDALS OR 0 OR 0 AND PUADDR |

**Byte (hex)**

**Contents**

**00–03** Record ID:

**C"ALSA"**

Add an adjacent link station.

**C"ALSD"**

Delete an adjacent link station.

**C"ALSP"**

Search for an APPN link station in the cross-domain resource's ALS list.

**C"ALSR"**

Replace an adjacent link station.

**C"ALSS"**

Select an adjacent link station.

**C"ALSV"**

Validate an adjacent link station.

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code from ALSLIST macro invocation

**06** Session control block identifier or 0

**07** Flags (from the following list):

| **Bit** | **Meaning** |
|---|---|
| **1... ....** | The type of add is dynamic. |
| **.1.. ....** | Autologon processing requested. |
| **..1. ....** | Verify that the add is required. |
| **...1 ....** | Waiting autologon requests should be processed. |
| **.... xx..** | 0 |
| **.... ..xx** | Connection type of PU whose address is given in PUADDR: |

| | |
|---|---|
| **00** | Unknown |
| **01** | LEN |
| **11** | APPN |

**08–0F** ALS name or 0
- For ALSA, ALSR, or ALSV: New ALS name passed on the add, replace, or validate function
- For ALSD or ALSP: 0
- For ALSS: ALS name returned on the select function

**10–13** Return address of the caller of the ALSLIST function

**14–17** Address of the cross-domain resource passed to the ALSLIST function

**18–1F** ALS name, PU address, or 0
- For ALSA or ALSP: 0
- For ALSD or ALSR: Old ALS name passed on the delete or replace function
- For ALSS or ALSV: 0 and PUADDR
  - 18–1B: 0

- 1C–1F: The PU address that is returned for the select or validate function

# AP entry for signals passed between components of the APPN CP

**Entry: AP**
**VIT option:**
  SSCP
**Event:** Signals passed between components of the APPN CP
**VIT processing module:**
  ISTRACSC
**Control is returned to:**
  Module invoking the INTRACE macro that caused the record to be produced.

This trace record provides information about inbound request and responses which are processed by the CP. If this entry is associated with an event failure, nonzero sense code, it is generated regardless of the SSCP option. It is recognized as an exception code and is traced when the VIT is active.

| 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 | 2 | 3 | 4 | 5 6 7 8 | 9 A B C | D E F 0 | 1 2 3 4 5 6 7 8 9 A B C D E F |
| AP | T O P A B | F R O M P A B | I D | MOD ID | RETURN ADDRESS | SENSE DATA OR 0 | FIRST 16 BYTES OF INTERPROCESS SIGNAL (IPS) |

**Byte (hex)**
  **Contents**
**00–01** Record ID: C"AP"
**02** To process anchor block (PAB) identification byte:

| Code (hex) | ID |
|---|---|
| "A" | Advanced program-to-program communication (APPC) |
| "B" | CMIP services session layer |
| "C" | Session services for CP-CP sessions (SSC) |
| "D" | APPN directory services (DR) |
| "E" | Directory services management exit (DSME) |
| "F" | CMIP services association control function, association control service element, directory service element, presentation |
| "G" | VTAM topology agent (TOPO) |
| "H" | Half session (HS) |
| "I" | Disk input or output |
| | **Note:** This code is an uppercase letter *I*. |
| "J" | Management services transport (MST) main processing |
| "K" | CMIP services initialization, termination, and command processing |
| "L" | Session services for LU-LU sessions (SSL) |
| "M" | Management services transport (MST) initialization processing |

| | | |
|---|---|---|
| "N" | Node operator services | |
| "O" | CMIP services management information base (MIB) controller | |
| "P" | SSCP functions | |
| "Q" | CMIP services scoping and replication | |
| "R" | CMIP services event filtering and forwarding | |
| "S" | System services control point (SSCP) | |
| "T" | Topology and routing services (TRS) | |
| "U" | Recovery RTP context manager (RCM) | |
| "V" | Recovery manager (RVM) | |
| "X" | Transaction programs (XP) | |
| "Y" | Recovery PAB (RPAB) | |
| "Z" | Dependent LU server (DLUS) | |
| "l" | LU server (LUS) | |

**Note:** This code is a lowercase letter *L*.

**03** From process anchor block (PAB) Identification byte. The codes are the same as byte 2.

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** Last three characters of the module invoking APSEND. (The characters *KWN* indicate that the module name is not known.)

**08–0B** Address of the issuer of the APSEND module

**0C–0F** Sense code or 0

**10–1F** First 16 bytes of the interprocess signal (IPS). For a list of interprocess signals, refer to *z/OS Communications Server: SNA Data Areas Volume 2*.

## A2 entry for requests and responses with a RUPE

**Entry: A2**
**VIT option:**
    SSCP
**Event:** Requests and Responses with a RUPE
**VIT processing module:**
    ISTRACSC

This trace record is a continuation of the AP entry.

| 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 | 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| A2 | 30 MORE BYTES OF THE INTERPROCESS SIGNAL (IPS) |

**Byte (hex)**
    **Contents**
**00–01** Record ID: C"A2"
**02–1F** 30 more bytes of the interprocess signal

## ARB entry for RTP LU-LU session data

**Entry: ARB**
**VIT option:**
    HPR
**Event:** Any LU-LU session data sent across an RTP connection

**VIT processing module:**
ISTITCHP
**Control is returned to:**
ISTRPCRS

This trace record is written when LU-LU session data is sent across an RTP connection. It contains statistical information on the LU-LU session data.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| ARB | I D | 0 | I N S T A N C E | F L A G S | SEND RATE | NEW BURST INTERVAL | BYTES SENT | BYTES TO SEND | PACKET SIZE | RPNCB ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"ARB"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Trace instance
**07** Information Flags and Burst Multiplier

| Bit | Meaning |
|---|---|
| **xx.. ....** | Adaptive Rate Based Algorithm Used |
| | **00** ARB Mode Algorithm |
| | **01** ARB Responsive Mode Algorithm |
| **..xx xxxx** | Burst Multiplier (RPN_BURST_MULT) |

**08–0B** Send rate in bytes per second
**0C–0F** Burst interval in milliseconds
**10–13** Bytes sent in the burst interval (RPN_BYTES_SENT)
**14–17** Bytes left to send (RPN_BYTES_TO_SEND)
**18–1B** Size of the network layer packet (NLP)
**1C–1F** RPNCB address

# ARBB entry for HPR trace option

**Entry: ARBB**
**VIT option:**
HPR
**Event:** All RTP connections using Responsive Mode ARB and ARBB active.
**VIT processing module:**
ISTITCHA
**Control is returned to:**
The module that issued the INTRACE macro.

This trace record is written when RTP connections are using Responsive Mode ARB and ARBB option is active. The trace record has six instances included as follows.

**Instance 1**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBB | | | | ID | 0 | INSTANCE | 0 | RATE | LAST RATE | 0 | 0 | DELAY CHANGE SUM | | | | CURRENT DELAY CHANGE | | | | ELAPSED TIME | | | | SMI | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"ARBB"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  0

**06**  Trace instance

**07**  0

**08**  ARB rate status (MRP_NLH_RATE)

**09**  Rate of last ARB measurement sent (ARB_LAST_RATE)

**0A–0B**  0

**0C–0F**  Delay Change Sum (DCS) — accumulated network delay in milliseconds (ARB_ACCUM_QTIME)

**10–13**  Current delay change (ARBLV_CURR_QTIME)

**14–17**  Elapsed time in milliseconds since the last ARB rate request (ARBLV_ETIME_IN_MS)

**18–1B**  ARB request sender's measurement interval (SMI) in microseconds (THD_ARB_CURR_MEASURE_INT)

**1C–1F**  RPNCB address

**Instance 2**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBB | | | | ID | 0 | INSTANCE | 0 | RATE | LAST RATE | 0 | 0 | DELAY CHANGE SUM | | | | DCS BOUNDARY REGION 1 AND 2 | | | | DCS BOUNDARY REGION 2 AND 3 | | | | DCS BOUNDARY REGION 3 AND 4 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"ARBB"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

| | |
|---|---|
| **05** | 0 |
| **06** | Trace instance |
| **07** | 0 |
| **08** | MRP_RCV_RATE_STAT |
| **09** | Rate of last ARB measurement sent (ARB_LAST_RATE) |
| **0A–0B** | 0 |
| **0C–0F** | Delay Change Sum (DCS) — accumulated network delay in milliseconds (ARB_ACCUM_QTIME) |
| **10–13** | DCS boundary in milliseconds dividing region 1 and 2 (ARB_GREEN_THRESHOLD) |
| **14–17** | DCS boundary in milliseconds dividing region 2 and 3 (ARB_MAX_QTIME_RETN ) |
| **18–1B** | DCS boundary in milliseconds dividing region 3 and 4 (ARB_MAX_QTIME_RETN * ACM_UPPER_LIM) |
| **1C–1F** | RPNCB address |

**Note:** The following ARB rate replies are returned to the sender of the ARB rate request depending on the region in which the DCS lies:

| DCS in region | Rate reply returned |
|---|---|
| 1 | NORMAL |
| 2 | RESTRAINT |
| 3 | SLOWDOWN 1 or SLOWDOWN 2 |
| 4 | RESTRAINT |

**Instance 3**

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBB | ID | 0 | INSTANCE | FLAGS | RATE CUT | FLAGS | 0 | 0 | ALLOWED SENDING RATE | INCREMENT | INCREMENT TREND | DECREMENT TREND | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

| | |
|---|---|
| **00–03** | Record ID: C"ARBB" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Trace instance |
| **07** | ARB flags passed on the ARB segment (THD_ARB_FLAGS) |
| **08** | Indicates size of rate cut for a SLOWDOWN (send_rate = send_rate — (send_rate / acm_cut_rate)) (ACM_CUT_RATE) |
| **09** | Flag values used for ARB processing (ARB_FLAGS) |
| **0A–0B** | 0 |
| **0C–0F** | ARB allowed sending rate in Kbps (ARB_ALLOW_SEND_RATE) |

**10–13**   Current increment value for allowed send rate in Kbps
(ARB_CURR_RATE_INC)
**14–17**   Number of consecutive sending rate increments (ARB_INC_TREND)
**18–1B**   Number of consecutive sending rate decrements (ARB_DEC_TREND)
**1C–1F**   RPNCB address

**Instance 4**

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
| 0 1 2 3 | 4 5 | 6 7 | 8 9 | A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|
| ARBB | I D | 0 | I N S T A N C E | 0 | F L A G S | 0 | 0 | 0 | FLAGS | INCREMENT TREND | DECREMENT TREND | RECEIVE RATE | RPNCB ADDRESS |

**Byte (hex)**

   **Contents**
**00–03**   Record ID: C"ARBB"
**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is
greater than X'FF'.
**05**   0
**06**   Trace instance
**07**   0
**08**   Flag values used for ARB processing (ARB_FLAGS)
**09**   0
**0A–0B**   0
**0C–0F**   Flag values used for ARB processing in Kbps (ARB_CURR_RATE_INC)
**10–13**   Number of consecutive sending rate increments (ARB_INC_TREND)
**14–17**   Number of consecutive sending rate decrements (ARB_DEC_TREND)
**18–1B**   Receiving rate of the sender in Kbps (THD_ARB_REC_RATE)
**1C–1F**   RPNCB address

**Instance 5**

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
| 0 1 2 3 | 4 5 | 6 7 | 8 9 | A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|
| ARBB | I D | 0 | I N S T A N C E | 0 | ALLOWED SEND RATE | MAX BURST | MIN BURST | MIN INCREMENT | MAX SEND RATE | RPNCB ADDRESS |

**Byte (hex)**
   **Contents**
00–03  Record ID: C"ARBB"
04     ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05     0
06     Trace instance
07     0
08–0B  ARB allowed sending rate in Kbps (ARB_ALLOW_SEND_RATE)
0C–0F  Maximum burst time allowed in milliseconds (ARB_BT_MAX)
10–13  Minimum burst time allowed in milliseconds (ARB_BT_MIN)
14–17  Minimum increment value for allowed send rate in Kbps (ARB_MIN_RATE_INC)
18–1B  Shared link rate in Kbps (ARB_MAX_SEND_RATE)
1C–1F  RPNCB address

**Instance 6**

| 0 0 0 0 <br> 0 1 2 3 | 0 0 <br> 4 5 | 0 <br> 6 | 0 <br> 7 | 0 0 0 0 <br> 8 9 A B | 0 0 0 0 <br> C D E F | 1 1 1 1 <br> 0 1 2 3 | 1 1 1 1 <br> 4 5 6 7 | 1 1 1 1 <br> 8 9 A B | 1 1 1 1 <br> C D E F |
|---|---|---|---|---|---|---|---|---|---|
| ARBB | I D | 0 I N S T A N C E | 0 | SMOOTH ROUND TRIP | SHORT REQUEST TIMER | ADJUST INTERVAL | MEASURE INTERVAL | 0 | RPNCB ADDRESS |

**Byte (hex)**
   **Contents**
00–03  Record ID: C"ARBB"
04     ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05     0
06     Trace instance
07     0
08–0B  Smoothed Round Trip Time in milliseconds (RPN_SRTTs)
0C–0F  Short Request timer in milliseconds (RPN_SHORT_REQ_T)
10–13  Counter for adjustment of Base ARB measurement interval (ARB_TIME_ADJUST_M_INT)
14–17  Interval in milliseconds for ARB measurement requests (ARB_MEASURE_INT)
18–1B  0
1C–1F  RPNCB address

# ARBR entry for HPR trace option

**Entry: ARBR**
**VIT option:**
   HPR
**Event:** All RTP connections using Responsive Mode ARB and ARBR active.

**VIT processing module:**
ISTITCHA
**Control is returned to:**
The module that issued the INTRACE macro.

This trace record is written when RTP connections are using Responsive Mode ARB and ARBR option is active. The trace record has 38 instances included as follows.

**Instance 0**

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | I D | 0 | I N S T A N C E | 0 | F C U R V E | R A T E | | DRAIN | | | | ALLOWED SEND RATE | | | | TARGET RATE | | | | LOSS BUCKET CONTENTS | | | | RPNCB ADDRESS | | | | | | |

**Byte (hex)**
**Contents**
00–03  Record ID: C"ARBR"
04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05  0
06  Trace instance
07  0
08–09  Aggressiveness indicator, the range is 0 (most aggressive) – 1000 (least aggressive) (ARB2_FCURVE_NOW)
0A–0B  Determines how fast F_curve value increases (ARB2_ALPHA_NOW)
0C–0F  Number of bytes drained from leaky bucket during last interval (DRAIN_IN_BYTES)
10–13  ARB allowed sending rate in Kbps (ARB_ALLOW_SEND_RATE)
14–17  Target sending rate in Kbps (ARB2_TARGET_RATE)
18–1B  Number of bytes that can be placed in the leaky bucket (ARB2_LOSS_BUCKET_CONTENTS)
1C–1F  RPNCB address

**Instance 1**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | CORRELATOR | CORRELATOR | | CORRELATOR | | CORRELATOR | | RECEIVER THRESHOLD | | | | MIN RECEIVER THRESHOLD | | | | MAX RECEIVER THRESHOLD | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

00–03   Record ID: C"ARBR"

04   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05   0

06   Trace instance

07   0

08   ARB flags passed on the ARB segment (THD_ARB_FLAGS)

09   ARB request correlator sent on ARB segment (THD_ARB_CORRELATOR)

0A–0B   Current ARB request correlator (ARB2_REQUEST_CORRELATOR)

0C–0D
   Local request correlator working field (ARBLV_REQ_CORRELATOR)

0E–0F   Previous ARB request correlator (ARB2_LAST_REQ_CORRELATOR)

10–13   Current receiver threshold in microseconds (ARB2_RCVR_THRESHOLD)

14–17   Minimum Receiver threshold in microseconds (ARB2_RCVR_THRESHOLD_MIN)

18–1B   Maximum Receiver threshold in microseconds (ARB2_RCVR_THRESHOLD_MAX)

1C–1F   RPNCB address

**Instance 2**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | 0 | 0 | 0 | 0 | CORRELATOR | | PARITY | PARITY | CORRELATOR | | | | 0 | | | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

00–03   Record ID: C"ARBR"

04   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05   0

| | |
|---|---|
| **06** | Trace instance |
| **07** | 0 |
| **08** | 0 |
| **09** | 0 |
| **0A–0B** | 0 |
| **0C–0D** | |
| | Previous ARB request correlator (ARB2_LAST_REQ_CORRELATOR) |
| **0E–0F** | Local ARB parity bit (ARBLV_TEMP_ARB_PARITY) |
| **10–11** | Local last ARB parity bit (ARBLV_TEMP_LAST_RCVD_PARITY) |
| **12–13** | Local request correlator working field (ARBLV_TEMP_REQ_CORRELATOR) |
| **14–17** | 0 |
| **18–1B** | 0 |
| **1C–1F** | RPNCB address |

**Instance 3**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | IDLE TIME | | | | CURRENT TIME | | | | LAST DATA RECEIVED | | | | IDLE TIME THRESHOLD | | | | MAX SEND RATE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| **00–03** | Record ID: C"ARBR" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Trace instance |
| **07** | 0 |
| **08–0B** | Amount of time idle in milliseconds since last data received (ARBLV_IDLE_TIME) |
| **0C–0F** | Current time stamp value (ARBLV_CURR_TIME) |
| **10–13** | Time stamp value of last data received (ARB2_LAST_DATA_RCVD) |
| **14–17** | Idle time in milliseconds threshold (INC_IDLE_TIME_THRESH(ARBLV_MAXSNDI)) |
| **18–1B** | Shared link rate table index (ARBLV_MAXSNDI) |
| **1C–1F** | RPNCB address |

**Instance 4**

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---------|---|---|---|---|---------|---------|---------|---------|---------|---------|---------|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | |
| ARBR | I D | 0 | I N S T A N C E | F L A G S | ELAPSED TIME | POTENTIAL DCS | POTENTIAL ERROR COUNT | DELAY CHANGE BAR | DC STANDARD DEVIATION | RPNCB ADDRESS |

**Byte (hex)**

   **Contents**
- **00–03** Record ID: C"ARBR"
- **04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
- **05** 0
- **06** Trace instance
- **07** Flag values used for ARB processing (ARB_FLAGS)
- **08–0B** Elapsed time in microseconds since last ARB status request (ARBLV_ETIME_IN_MS)
- **0C–0F** Potential DCS — accumulated network delay in microseconds (ARB2_POT_ACCUM_QTIME)
- **10–13** Potential Count of errors detected (ARB2_POT_ERROR_COUNT)
- **14–17** Delay Change bar value in microseconds (ARB2_DC_BAR)
- **18–1B** Delay Change standard deviation in microseconds (ARB2_DC_SDEV)
- **1C–1F** RPNCB address

**Instance 5**

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---------|---|---|---|---|---------|---------|---------|---------|---------|---------|---------|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | |
| ARBR | I D | 0 | I N S T A N C E | 0 | POT - PC BAR VALUE | POT - PC STANDARD DEVIATION | POT - PC THRESHOLD | AVERAGE COEFFICIENT | DC STANDARD DEVIATION | RPNCB ADDRESS |

**Byte (hex)**

   **Contents**
- **00–03** Record ID: C"ARBR"
- **04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
- **05** 0
- **06** Trace instance
- **07** 0
- **08–0B** ARB2_POT_DC_BAR (Potential Delay Change bar value in microseconds)

**0C–0F** ARB2_POT_DC_SDEV (Potential Delay Change standard deviation in microseconds)
**10–13** ARB2_POT_DC_THRESH (Potential Delay Change threshold in microseconds)
**14–17** ARBLV_AVG_COEFF (Average Coefficient based on elapsed time between ARB status requests)
**18–1B** ARB2_DC_SDEV (Delay Change standard deviation in microseconds)
**1C–1F** RPNCB address

**Instance 6**

| 0 0 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| ARBR | ID | 0 | INSTANCE | FLAGS  DCS BAR VALUE | AVG COEFFICIENT | POTENTIAL DCS | DCS SQUARED BAR | MAX DC | RPNCB ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"ARBR"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Trace instance
**07** ARB_FLAGS (Flag values used for ARB processing)
**08–0B** ARB2_DCS_BAR (Delay Change Sum bar value in microseconds)
**0C–0F** ARBLV_AVG_COEFF (Average Coeffecient based on elapsed time between ARB status requests)
**10–13** ARB2_POT_ACCUM_QTIME (Potential DCS — accumulated network delay in microseconds)
**14–17** ARB2_DCS_SQUARED_BAR (Delay Change Sum bar value squared in milliseconds)
**18–1B** ARB2_ACCUM_QTIME_MAX (Maximum Delay Change over last 30 ARB status requests in microseconds)
**1C–1F** RPNCB address

**Instance 7**

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | ID | 0 | INSTANCE | 0 | DCS | DC THRESHOLD | 0 | 0 | 0 | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

| 00–03 | Record ID: C"ARBR" |
|---|---|
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | Trace instance |
| 07 | 0 |
| 08–0B | Delay Change Sum(DCS) — accumulated network delay in microseconds (ARBLV_CURR_QTIME) |
| 0C–0F | Delay Change threshold in microseconds (ARB2_DC_THRESH) |
| 10–13 | 0 |
| 14–17 | 0 |
| 18–1B | 0 |
| 1C–1F | RPNCB address |

**Instance 8**

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | ID | 0 | INSTANCE | FLAGS | POTENTIAL DCS BAR | POTENTIAL DCS SQUARED BAR | DCS VARIANCE | MAX DCS | MAX DC | RPNCB ADRESS |

**Byte (hex)**

**Contents**

| 00–03 | Record ID: C"ARBR" |
|---|---|
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | Trace instance |
| 07 | Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS) |

**08–0B**  Potential Delay Change Sum(DCS) bar value in microseconds
(ARB2_POT_DCS_BAR)
**0C–0F**  Potential DCS bar value squared in milliseconds
(ARB2_POT_DCS_SQUARED_BAR)
**10–13**  Delay Change Sum variance (ARBLV_DCS_VARIANCE)
**14–17**  Maximum Delay Change Sum recorded in microseconds
(ARB2_DCS_MAX_OVERALL)
**18–1B**  Maximum Delay Change in microseconds over last 30 ARB status requests
(ARB2_ACCUM_QTIME_MAX)
**1C–1F**  RPNCB address

**Instance 9**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | FLAGS | FLAGS | POTENTIAL DCS | | | | LAST DCS | | | | RECEIVER THRESHOLD | | | | MAX SEND RATE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"ARBR"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  0
**06**  Trace instance
**07**  0
**08**  Flag values used for ARB processing (ARB_FLAGS)
**09**  Flag values used for Responsive Mode ARB processing
(ARB2_PHASE_FLAGS)
**0A**  Potential flag values used for Responsive Mode ARB processing
(ARB2_POT_PHASE_FLAGS)
**0B**  Global ARB segment flags (MRP_ARB_FLAGS)
**0C–0F**  Potential DCS — accumulated network delay in milliseconds
(ARB2_POT_ACCUM_QTIME)
**10–13**  Last Delay Change Sum(DCS) in microseconds
(ARBLV_LAST_ACCUM_QTIME)
**14–17**  Receiver Threshold value in microseconds (ARB2_RCVR_THRESHOLD)
**18–1B**  Shared link rate in Kbps (ARB_MAX_SEND_RATE)
**1C–1F**  RPNCB address

**Instance 10**

| 0 0 | 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | FLAGS | FLAGS | SMOOTH SEND RATE | ELAPSED TIME | POTENTIAL MAX | MAX DELAY | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08** Flag values used for ARB processing (ARB_FLAGS)

**09** Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS)

**0A** Potential flag values used for Responsive Mode ARB processing (ARB2_POT_PHASE_FLAGS)

**0B** Global ARB segment flags (MRP_ARB_FLAGS)

**0C–0F** Smoothed sending rate in Kbps (ARB2_SMOOTHED_RATE)

**10–13** Elapsed time in microseconds since last ARB status request (ARBLV_ETIME_IN_MS)

**14–17** Potential maximum (ARB2_POT_MAX_DCS_WINDOW)

**18–1B** Maximum Delay in microseconds Change Sum recorded (ARB2_POT_DCS_MAX_OVERALL)

**1C–1F** RPNCB address

**Instance 11**

| 0 0 | 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | ID | 0 | INSTANCE | 0 | LAST CURRENT BYTE RECEIVED | | | | BYTE RECEIVED LAST ARB | BITS OF DATA – LAST INTERVAL | POTENTIAL SMOOTHED RATE | MAX WINDOW | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

| | |
|---|---|
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Trace instance |
| **07** | 0 |
| **08–0B** | Last current byte received (RPN_LAST_BYTE_RCV) |
| **0C–0F** | Byte received during last ARB status request (ARB2_LAST_BYTE_RCV) |
| **10–13** | Bits of data received over last interval (ARBLV_BITS_OF_DATA) |
| **14–17** | Potential smoothed sending rate in Kbps (ARB2_POT_SMOOTHED_RATE) |
| **18–1B** | Maximum window size over last 10 ARB status requests (ARB2_POT_MAX_DCS_WINDOW) |
| **1C–1F** | RPNCB address |

**Instance 12**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | POTENTIAL DCS BAR | | | | THRESHOLD MAX | | | | THRESHOLD VARIANCE | | | | INCREMENTS TO WINDOW SIZE | | | | INCREMENTS OVER TIME | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| **00–03** | Record ID: C"ARBR" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Trace instance |
| **07** | 0 |
| **08–0B** | Potential Delay Change Sum bar value in microseconds (ARB2_POT_DCS_BAR) |
| **0C–0F** | Present Receiver Threshold maximum in microseconds (ARB2_PRESENT_THR_MAX) |
| **10–13** | Computed threshold variance in microseconds for this link (ARB2_ANTICIPATED_VAR) |
| **14–17** | Fraction that indicates number of increments related to window size (value is 0–1000) (ARB2_N_FRACTION) |
| **18–1B** | Fraction that indicates number of increments over time (value is 0–100) (ARB2_N_TREND) |
| **1C–1F** | RPNCB address |

**Instance 13**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | RECOVERY FACTOR | | | | POTENTIAL DCS BAR | | | | ORIGINAL MAX THRESHOLD | | | | POTENTIAL MAX THRESHOLD | | | | POTENTIAL THRESHOLD VARIANCE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Dictates how long the competor stays active (values 0–100) (ARB2_RECOVERY_FACTOR)

**0C–0F** Potential Delay Change Sum bar value in microseconds (ARB2_POT_DCS_BAR)

**10–13** Original Threshold maximum in microseconds (ARB2_ORIG_THR_MAX)

**14–17** Potential Present Receiver Threshold maximum in microseconds (ARB2_POT_PRESENT_THR_MAX)

**18–1B** Potential threshold variance in microseconds for this link (ARB2_POT_ANTICIPATED_VAR)

**1C–1F** RPNCB address

**Instance 14**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | RECOVERY FACTOR | | | | POTENTIAL MIN THRESHOLD | | | | POTENTIAL MAX THRESHOLD | | | | POTENTIAL THRESHOLD VARIANCE | | | | ORIGINAL THRESHOLD MAX | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0
**08–0B** Dictates how long the competor stays active (values 0–100)
(ARB2_RECOVERY_FACTOR)
**0C–0F** Potential Present Receiver Threshold minimum in microseconds
(ARB2_POT_PRESENT_THR_MIN)
**10–13** Potential Present Receiver Threshold maximum in microseconds
(ARB2_POT_PRESENT_THR_MAX)
**14–17** Potential threshold variance in microseconds for this link
(ARB2_POT_ANTICIPATED_VAR)
**18–1B** Original Threshold maximum in microseconds (ARB2_ORIG_THR_MAX)
**1C–1F** RPNCB address

**Instance 15**

| 0 0 0 0<br>0 1 2 3 | 0<br>4 | 0<br>5 | 0<br>6 | 0<br>7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1 1 1<br>0 1 2 3 | 1 1 1 1<br>4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | I D | 0 | I N S T A N C E | 0 | POTENTIAL MIN THRESHOLD | POTENTIAL MAX THRESHOLD | POTENTIAL THRESHOLD VARIANCE | SAVED MIN THRESHOLD | SAVED MAX THRESHOLD | RPNCB ADDRESS |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"ARBR"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is
greater than X'FF'.
**05** 0
**06** Trace instance
**07** 0
**08–0B** Potential Present Receiver Threshold minimum in microseconds
(ARB2_POT_PRESENT_THR_MIN)
**0C–0F** Potential Present Receiver Threshold maximum in microseconds
(sARB2_POT_PRESENT_THR_MAX)
**10–13** Potential threshold variance in microseconds for this link
(ARB2_POT_ANTICIPATED_VAR)
**14–17** Saved Receiver Threshold minimum in microseconds
(ARB2_SAVED_THR_MIN)
**18–1B** Saved Receiver Threshold maximum in microseconds
(ARB2_SAVED_THR_MAX)
**1C–1F** RPNCB address

**Instance 16**

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | ID | 0 | I N S T A N C E | 0 | WINDOW SIZE | INCREMENTS TO WINDOW SIZE | NUMBER INCREMENTS | LAST N_FRACTION | INCREMENTS OVER TIME | RPNCB ADDRESS |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Number of ARB status requests used in calculating Receiver thresholds (ARB2_WINDOW_SIZE)

**0C–0F** Fraction that indicates number of increments related to window size (value 0–1000) (ARB2_N_FRACTION)

**10–13** Number of consecutive sending rate increments (ARB_INC_TREND)

**14–17** Last N_Fraction (value is 0–1000) (ARBLV_N_FRACTION_LAST)

**18–1B** Fraction that indicates number of increments over time (value is 0–100) (ARB2_N_TREND)

**1C–1F** RPNCB address

**Instance 17**

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | ID | 0 | I N S T A N C E | 0 | WINDOW SIZE | NUMBER INCREMENTS | INCREMENTS OVER TIME | POTENTIAL MAX THRESHOLD | ORIGINAL MAX THRESHOLD | RPNCB ADDRESS |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Number of ARB status requests used in calculating Receiver thresholds (ARB2_WINDOW_SIZE)

**0C–0F** Number of consecutive sending rate increments (ARB_INC_TREND)

**10–13** Fraction that indicates number of increments over time (value 0–100) (ARB2_N_TREND)

**14–17** Potential Present Receiver Threshold maximum in microseconds (ARB2_POT_PRESENT_THR_MAX)

**18–1B** Original Threshold maximum in microseconds (ARB2_ORIG_THR_MAX)

**1C–1F** RPNCB address

**Instance 18**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | | 0 | INSTANCE | 0 | POTENTIAL RECEIVER THRESHOLD | | | | MIN RECEIVER THRESHOLD | | | | MAX RECEIVER THRESHOLD | | | | POTENTIAL MAX THRESHOLD | | | | ORIGINAL MAX THRESHOLD | | | | RPNCB ADDRESS | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Potential Receiver Threshold value in microseconds (ARB2_POT_RCVR_THRESHOLD)

**0C–0F** Minimum Receiver Threshold value in microseconds (ARB2_RCVR_THRESHOLD_MIN)

**10–13** Maximum Receiver Threshold value in microseconds (ARB2_RCVR_THRESHOLD_MAX)

**14–17** Potential Present Threshold maximum in microseconds (ARB2_POT_PRESENT_THR_MAX)

**18–1B** Original Threshold maximum in microseconds (ARB2_ORIG_THR_MAX)

**1C–1F** RPNCB address

**Instance 19**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | CORRELATOR | FLAGS | CORRELATOR | | MAX SEND RATE | | | | ALLOWED SEND RATE | | | | TARGET SEND RATE | | | | SMOOTH SEND RATE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08** ARB request correlator sent on ARB segment (THD_ARB_CORRELATOR)

**09** Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS2)

**0A–0B** Current ARB request correlator (ARB2_REQUEST_CORRELATOR)

**0C–0F** Shared link rate in Kbps (ARB_MAX_SEND_RATE)

**10–13** ARB allowed sending rate in Kbps (ARB_ALLOW_SEND_RATE)

**14–17** Target sending rate in Kbps (ARB2_TARGET_RATE)

**18–1B** Smoothed actual sending rate in Kbps (ARB2_SMOOTH_ACTUAL_RATE)

**1C–1F** RPNCB address

**Instance 20**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | SMOOTH ROUND TRIP | | | | BURST | | FCURVE | | RATE | | SND/TRGT | | NUM STEPS | | 0 | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Smoothed Round Trip Time in milliseconds (RPN_SRTT)

**0C–0D**

ARB Burst interval in milliseconds (ARB2_BURST_INTVL)

**0E–0F** Aggressiveness indicator, range is 0 (most aggressive) – 1000 (least aggressive) (ARB2_FCURVE_NOW)

**10–11** Determines how fast F_curve value increases (ARB2_ALPHA_NOW)

**12–13** Fraction of the current send/target rate (value 0-1000) (ARB2_DOWN_GAMMA)

**14–15** Number of aggressive steps taken (ARB2_AGGRESSIVE_STEPS)

**16–17** 0

**18–1B** 0

**1C–1F** RPNCB address

**Instance 21**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | I D | 0 | I N S T A N C E | 0 | C O R R E L A T O R | F L A G S | 0 | 0 | C O R R E L A T O R | | 0 | | 0 | | | | 0 | | | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08** ARB request correlator sent on ARB segment (THD_ARB_CORRELATOR)

**09** Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS2)

**0A** 0

**0B** 0

**0C–0D**

Current ARB request correlator (ARB2_REQUEST_CORRELATOR)

**0E–0F** 0

**10–13** 0

**14–17** 0

**18–1B** 0

**1C–1F** RPNCB address

**Instance 22**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| ARBR | | | | ID | 0 | INSTANCE | 0 | 0 | | | | ALLOWED SEND RATE | | | | TARGET SEND RATE | | | | SLOW CNT | | CON CNT | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** 0

**0C–0F** ARB allowed sending rate in Kbps (ARB_ALLOW_SEND_RATE)

**10–13** Target sending rate in Kbps (ARB2_TARGET_RATE)

**14–15** Number of consecutive slowdowns while utilizing above 10% of shared link rate (ARB2_SLOW_COUNT)

**16–17** Number of consecutive ARB status requests while in conservative phase (ARB2_CONSERVATIVE_CNT)

**18–1B** 0

**1C–1F** RPNCB address

**Instance 23**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | FCURVE | | BURST SIZE | | | | MIN BURST SIZE | | | | RATE CHANGE | | | | ELAPSED TIME | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08** Flag values used for ARB processing (ARB_FLAGS)

**09** ARB flags passed on the ARB segment (THD_ARB_FLAGS)

**0A–0B** Aggressiveness indicator, range is 0 (most aggressive) – 1000 (least aggressive) (ARB2_FCURVE_NOW)

**0C–0F** Number of bits in burst size based on link rate (ARBLV_BURST_SIZE_MIN)

**10–13** Current minimum burst size (ARB2_BURST_SIZE_MIN)

**14–17** Fractional value used to determine rate increase or decrease (ARBLV_GAMMA_VALUE)

**18–1B** Elapsed time in milliseconds since last ARB status reply received (ARBLV_ETIME_IN_MS)

**1C–1F** RPNCB address

**Instance 24**

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| ARBR | ID | 0 | INSTANCE 0 | BITS OF DATA | BURST SIZE | CURRENT SEND RATE | RATE CHANGE | MIN GAMMA | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Bits of data sent over the last interval (ARBLV_BITS_OF_DATA)

**0C–0F** Number of bits in burst size based on link rate (ARBLV_BURST_SIZE_MIN)

**10–13** Current sending rate of last interval (ARBLV_CURRENT_SEND_RATE)

**14–17** Fractional value used to determine rate increase or decrease (ARBLV_GAMMA_VALUE)

**18–1B** Minimum Gamma value (ARBLV_GAMMA_MIN)

**1C–1F** RPNCB address

**Instance 25**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | HIGH STEP | | NEXT BYTE | | | | LAST BYTE SENT | | | | SMOOTH SEND RATE | | | | MAX SEND RATE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08** Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS)

**09** Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS2)

**0A–0B** Number of instances where allowed sending rate is above shared link rate (ARB2_HIGH_STEPS)

**0C–0F** Sequence number of next byte to be transmitted (RPN_NEXT_BYTE_XMIT)

**10–13** Last byte transmitted during previous ARB rate reply (ARB2_LAST_BYTE_SENT)

**14–17** Smoothed actual sending rate in Kbps (ARB2_SMOOTH_ACTUAL_RATE)

**18–1B** Shared link rate in Kbps (ARB_MAX_SEND_RATE)

**1C–1F** RPNCB address

**Instance 26**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | INDEX | 0 | HIGH STEP | | 0 | | MAX SEND RATE | | | | SMOOTH SEND RATE | | | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**     0
**06**     Trace instance
**07**     0
**08**     Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS)
**09**     Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS2)
**0A**     Shared link rate table index (ARBLV_MAXSNDI)
**0B**     0
**0C–0D**
    Number of instances where allowed sending rate is above ″shared link rate″ (ARB2_HIGH_STEPS)
**0E–0F**     0
**10–13**     Shared link rate in Kbps (ARB_MAX_SEND_RATE)
**14–17**     Smoothed actual sending rate in Kbps (ARB2_SMOOTH_ACTUAL_RATE)
**18–1B**     0
**1C–1F**     RPNCB address

**Instance 27**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | NUM STEP | | HIGH STEP | | SLOW COUNT | | CON COUNT | | FCURVE | | 0 | 0 | GAMMA VALUE | | | | SEND TARGET | | | | RPNCB ADDRESS | | | |

**Byte (hex)**
    **Contents**
**00–03**     Record ID: C″ARBR″
**04**     ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**     0
**06**     Trace instance
**07**     0
**08–09**     Number of aggressive steps taken (ARB2_AGGRESSIVE_STEPS)
**0A–0B**     Number of instances where allowed sending rate is above ″shared link rate″ (ARB2_HIGH_STEPS)
**0C–0D**
    Number of consecutive slowdowns while utilizing above 10% of shared link rate (ARB2_SLOW_COUNT)
**0E–0F**     Number of consecutive ARB status requests while in conservative phase (ARB2_CONSERVATIVE_CNT)
**10–11**     Aggressiveness indicator, range is 0 (most aggressive) – 1000 (least aggressive) (ARB2_FCURVE_NOW)
**12–13**     0
**14–17**     Fractional value used to determine rate increase or decrease (ARBLV_GAMMA_VALUE)

18–1B  Fraction of the current send/target rate (value 0 – 1000)
(ARB2_DOWN_GAMMA)

1C–1F  RPNCB address

**Instance 28**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | CON COUNT | | TARGET SEND RATE | | | | CURRENT SEND RATE | | | | MAX SEND RATE | | | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

00–03  Record ID: C"ARBR"

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05  0

06  Trace instance

07  0

08  Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS)

09  Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS2)

0A–0B  Number of consecutive ARB status requests while in conservative phase (ARB2_CONSERVATIVE_CNT)

0C–0F  Target for sending rate in Kbps (ARBLV_TARGET_RATE)

10–13  Current sending rate in Kbps (ARBLV_CURR_RATE)

14–17  Shared link rate in Kbps (ARB_MAX_SEND_RATE)

18–1B  0

1C–1F  RPNCB address

**Instance 29**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | CON COUNT | | | | 0 | | | | RANDOM VALUE | | | | SMOOTH ROUND TRIP | | | | SMOOTH ROUND TRIP LOWER BOUND | | | | RANDOM THRESHOLD | | RPNCB ADDRESS | |

**Byte (hex)**

**Contents**

| Byte (hex) | Contents |
|---|---|
| 00–03 | Record ID: C"ARBR" |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | Trace instance |
| 07 | 0 |
| 08–09 | Number of consecutive ARB status requests while in conservative phase (ARB2_CONSERVATIVE_CNT) |
| 0A–0B | 0 |
| 0C–0F | Random value in the range 0–16 (ITSUDATA) |
| 10–13 | Smoothed Round Trip Time in milliseconds (RPN_SRTT) |
| 14–17 | Smoothed Round Trip Time with a lower bound of 100 milliseconds (ARBLV_X) |
| 18–1B | Random Threshold value used to determine when to leave conservative phase (ARB2_RANDOM_THRESH) |
| 1C–1F | RPNCB address |

**Instance 30**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | FLAGS | FCURVE | | RATE | | SLOW COUNT | | TARGET SEND RATE | | | | CURRENT SEND RATE | | | | GAMMA VALUE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

| Byte (hex) | Contents |
|---|---|
| 00–03 | Record ID: C"ARBR" |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | Trace instance |
| 07 | 0 |
| 08 | Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS) |
| 09 | Flag values used for ARB processing (ARB_FLAGS) |
| 0A–0B | Aggressiveness indicator, range is 0 (most aggressive) – 1000 (least aggressive) (ARB2_FCURVE_NOW) |
| 0C–0D | Determines how fast F_curve value increases (ARB2_ALPHA_NOW) |
| 0E–0F | Number of consecutive slowdowns while utilizing above 10% of shared link rate (ARB2_SLOW_COUNT) |
| 10–13 | Target for sending rate in Kbps (ARBLV_TARGET_RATE) |
| 14–17 | Current sending rate in Kbps (ARBLV_CURR_RATE) |
| 18–1B | Fractional value used to determine rate increase or decrease (ARBLV_GAMMA_VALUE) |

1C–1F   RPNCB address

**Instance 31**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | I D | 0 | I N S T A N C E | 0 | 0 | | | | VARIABLE X | | | | VARIABLE B | | | | VARIABLE C | | | | SHARED RATE FRACTION | | | | RPNCB ADDRESS | | | |

**Byte (hex)**
> **Contents**

00–03   Record ID: C"ARBR"
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05      0
06      Trace instance
07      0
08–0B   0
0C–0F   Local variable used to compute decreased target rate (ARBLV_X)
10–13   Local variable used to compute decreased target rate (ARBLV_B)
14–17   Local variable used to compute decreased target rate (ARBLV_C)
18–1B   Fractional amount of shared link rate utilized (value 0-1000) (ARBLV_SHARED_RATE_FRACTION)
1C–1F   RPNCB address

**Instance 32**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | I D | 0 | I N S T A N C E | F L A G S | S L O W   C O U N T | | R A T E | | TARGET SEND RATE | | | | VARIABLE | | | | VARIABLE | | | | 0 | | | | RPNCB ADDRESS | | | |

**Byte (hex)**
> **Contents**

00–03   Record ID: C"ARBR"
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

| | | |
|---|---|---|
| **05** | 0 | |
| **06** | Trace instance | |
| **07** | Flag values used for Responsive Mode ARB processing (ARB2_PHASE_FLAGS) | |
| **08–09** | Number of consecutive slowdowns while utilizing above 10% of shared link rate (ARB2_SLOW_COUNT) | |
| **0A–0B** | Determines how fast F_curve value increases (ARB2_ALPHA_NOW) | |
| **0C–0F** | Target for sending rate in Kbps (ARBLV_TARGET_RATE) | |
| **10–13** | Local variable used to compute decreased target rate (ARBLV_B) | |
| **14–17** | Local variable used to compute decreased target rate (ARBLV_C) | |
| **18–1B** | 0 | |
| **1C–1F** | RPNCB address | |

**Instance 33**

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|
| ARBR | ID  0 | INSTANCE  0 | MIN STEPS TO TARGET | MAX STEPS TO TARGET | NUMBER STEPS TO TARGET | TRANS TIME FORWARD | TRANS TIME REVERSE | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

| | |
|---|---|
| **00–03** | Record ID: C"ARBR" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | Trace instance |
| **07** | 0 |
| **08–0B** | Minimum number of steps to reach target rate (multiplied by 1000) (ARBLV_KUP_MIN) |
| **0C–0F** | Maximum number of steps to reach target rate (multiplied by 1000) (ARBLV_KUP_MAX) |
| **10–13** | Number of steps to be taken to reach target rate (multiplied by 1000) (ARBLV_KUP_NOW) |
| **14–17** | Accumulated trans time forward in microseconds (ARB_ACCUM_TTIME_FRW) |
| **18–1B** | Accumulated trans time reverse in microseconds (ARB_ACCUM_TTIME_REV) |
| **1C–1F** | RPNCB address |

**Instance 34**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FLAGS | 0 | | | BURST INTERNAL | | | | NUMBER STEPS TO TARGET | | | | MIN BURST TIME | | | | NUMBER BITS BURST SIZE | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08** Flag values used for ARB processing (ARB_FLAGS)

**09–0B** 0

**0C–0F** Burst interval, which is the maximum of SRTT and 100 milliseconds (ARBLV_BURST_INTERVAL)

**10–13** Number of steps to be taken to reach target rate (multiplied by 1000) (ARBLV_KUP_NOW)

**14–17** Minimum ARB burst time duration in milliseconds (ARB_BT_MIN)

**18–1B** Number of bits in burst size based on link rate (ARBLV_BURST_SIZE_MIN)

**1C–1F** RPNCB address

**Instance 35**

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | SMOOTH ROUND TRIP | | | | SMOOTH DEVIATION | | | | SHORT REQUEST TIME | | | | CURRENT ROUND TRIP | | | | REFIFO TIME | | | | RPNCB ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–0B** Smoothed Round Trip Time in milliseconds (RPN_SRTT)

0C–0F  Smoothed deviation of Round Trip Time in milliseconds
       (RPN_SMOOTH_DEV)
10–13  Short Request time in milliseconds (RPN_SHORT_REQ_T)
14–17  Current Round Trip Time in milliseconds (ARB2_CURRENT_RTT)
18–1B  REFIFO time value in milliseconds (RPN_REFIFO_T)
1C–1F  RPNCB address

**Instance 36**

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | | | | ID | 0 | INSTANCE | 0 | FCURVE | | 0 | | IDLE TIME | | | | ALLOWED SEND RATE | | | | TARGET SEND RATE | | | | CURRENT ROUND TRIP | | | | RPNCB ADDRESS | | | |

**Byte (hex)**
       **Contents**
00–03  Record ID: C"ARBR"
04     ID is the primary address space ID (ASID). This field is 0 if the ASID is
       greater than X'FF'.
05     0
06     Trace instance
07     0
08–09  Aggressiveness indicator, range is 0 (most aggressive) – 1000 (least
       aggressive) (ARB2_FCURVE_NOW)
0A–0B  0
0C–0F  Amount of time in milliseconds idle since last data transmitted
       (ITSUDATA)
10–13  ARB allowed sending rate in Kbps (ARB_ALLOW_SEND_RATE)
14–17  Target sending rate in Kbps (ARB2_TARGET_RATE)
18–1B  Current Round Trip Time in milliseconds (ARB2_CURRENT_RTT)
1C–1F  RPNCB address

**Instance 37**

| 0 0 0 0 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 0 9 | 0 A 0 B | 0 C 0 D 0 E 0 F | 1 0 1 1 1 2 1 3 | 1 4 1 5 1 6 1 7 | 1 8 1 9 1 A 1 B | 1 C 1 D 1 E 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ARBR | ID | 0 | INSTANCE | 0 | BURST INT | FCURVE | MAX SEND RATE | SMOOTH ACTUAL SEND RATE | TARGET SEND RATE | NUMBER BITS BURST SIZE | RPNCB ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ARBR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace instance

**07** 0

**08–09** ARB Burst interval in milliseconds (ARB2_BURST_INTVL)

**0A–0B** Aggressiveness indicator, range is 0 (most aggressive) – 1000 (least aggressive) (ARB2_FCURVE_NOW)

**0C–0F** Shared link rate in Kbps (ARB_MAX_SEND_RATE)

**10–13** Smoothed actual sending rate in Kbps (ARB2_SMOOTH_ACTUAL_RATE)

**14–17** Target sending rate in Kbps (ARB2_TARGET_RATE)

**18–1B** Number of bits in burst size based on link rate (ARBLV_BURST_SIZE_MIN)

**1C–1F** RPNCB address

# AREL entry for abend RELSTORE

**Entry: AREL**

**VIT option:**
SMS

**Event:** Abend RELSTORE

**VIT processing module:**
ISTRACSM

**Control is returned to:**
ISTORAPR

This trace record identifies the buffers that are released by VTAM when a VTAM application program is terminated.

| 0 0 0 0 | 0 4 | 0 5 | 0 6 0 7 | 0 8 0 9 0 A 0 B | 0 C 0 D 0 E 0 F | 1 0 1 1 1 2 1 3 | 1 4 1 5 1 6 1 7 1 8 1 9 1 A 1 B 1 C 1 D 1 E 1 F |
|---|---|---|---|---|---|---|---|
| AREL | ID | CBID | 0 | PST ADDRESS | BUFFER ADDRESS | RETURN ADRESS | 0 |

**Byte (hex)**

**Contents**

00–03   Record ID: C"AREL"
04       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05       Control block ID index value
06–07   0
08–0B   PST address
0C–0F   Address of buffer being released
10–13   Address of the caller of the abend RELSTORE routine
14–1F   0

# ARP or ARS entry for RTP session data reply (Part 1)

**Entry:**  **ARP or ARS**
**VIT option:**
     HPR
**Event:**  Any session data reply is sent over an RTP connection using Responsive Mode ARB
**VIT processing module:**
     ISTITCHR
**Control is returned to:**
     ISTRPCAR

This trace record is written when a session data reply is sent across an RTP connection using Responsive Mode ARB, or when simulating a slowdown reply. It contains statistical information on the session data.

| 00 01 02 03 | 04 | 05 | 06 07 | 08 09 | 0A 0B | 0C 0D 0E 0F | 10 11 | 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| ARP OR ARS | I D | 0 | P H A S E F L A G S | H I G H S T E P S | A G G R E S S I V E | S T E P S | MEASURE INT | B U R S T I N T E R V A L | D O W N W A R D G A M M A V A L U E | GAMMA VALUE | NCB ADDRESS | RPH ADDRESS |

**Byte (hex)**
     **Contents**
00–03   Record ID:

- C"ARP" for real reply
- C"ARS" for simulated reply

04       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05       0
06–07   Phase flags (ARB2_PHASE_FLAGS, ARB2_PHASE_FLAGS2)

     **Bit**     **Meaning**
     **1... .... .... ....**
          Conservation phase

.1.. .... .... ....
   Medium aggressive phase
..1. .... .... ....
   Send setup segment
...1 .... .... ....
   High flag
.... 1... .... ....
   Delay change sum flag
.... .1.. .... ....
   Upward cycle flag
.... ..1. .... ....
   Idle - no data has been sent recently
.... ...1 .... ....
   Startup
.... .... 1... ....
   Last parity bit received on an ARB rate request
.... .... .1.. ....
   Current ARB request parity bit
.... .... ..11 1111
   Not used

| Byte (hex) | Contents |
|---|---|
| 08–09 | High steps (ARB2_HIGH_STEPS) – the number of successive normal rate increases above the current shared link rate |
| 0A–0B | Aggressive steps (ARB2_AGGRESSIVE_STEPS) |
| 0C–0F | Measurement interval in milliseconds (ARB2_MEASURE_INT) |
| 10–11 | Burst interval in milliseconds (ARB2_BURST_INTVL) |
| 12–13 | Downward gamma value (ARB2_DOWN_GAMMA) |
| 14–17 | Gamma value |
| 18–1B | NCB address |
| 1C–1F | Request parameter header (RPH) address |

# ARP2 or ARS2 entry for RTP session data reply (Part 2)

**Entry:** ARP2 or ARS2
**VIT option:**
   HPR
**Event:** Any session data reply is sent over an RTP connection using Responsive Mode ARB
**VIT processing module:**
   ISTITCHR

This trace record is a continuation of the ARP or ARS entry.

| 00 01 02 03 | 04 05 | 06 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|
| ARP2 OR ARS2 | CURRENT | CURRENT | CURRENT TRACK SPEED (Kup) | MAXIMUM SEND RATE | ALLOWED SEND RATE | TARGET RATE | SMOOTH ACTUAL RATE | SMOOTH ROUND TRIP TIME |
| | FCURVENT | ALPHREANT | | | | | | |

**Byte (hex)**
**Contents**

00–03  Record ID:
- C"ARP2" for real reply
- C"ARS2" for simulated reply

04–05  Current F-curve value (ARB2_FCURVE_NOW)
06–07  Current alpha value (ARB2_ALPHA_NOW)
08–0B  Current tracking speed value
0C–0F  Maximum send rate in Kbps (ARB_MAX_SEND_RATE)
10–13  Allowed send rate in Kbps (ARB_ALLOW_SEND_RATE)
14–17  Target rate in Kbps (ARB2_TARGET_RATE)
18–1B  Smooth actual rate in Kbps (ARB2_SMOOTH_ACTUAL_RATE)
1C–1F  Smooth round-trip time in milliseconds (RPN_SRTT)

# ARQ entry for RTP session data request (Part 1)

**Entry: ARQ**
**VIT option:**
        HPR
**Event:** Any session data request is received over an RTP connection using
        Responsive Mode ARB
**VIT processing module:**
        ISTITCHR
**Control is returned to:**
        ISTRPCAR

This trace record is written when session data is received across an RTP connection using Responsive Mode ARB. It contains statistical information on the session data.

| 00 01 02 | 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ARQ | INSTANCE | ID | 0 | PHASE OR WINDOW OR 0 | FLAGS OR SIZE | RECEIVR THRESH OR CLOCK DRIFT | MAXIMUM THRESH OR DRIFT INTERVAL | MINIMUM THRESH OR ANTICIP VAR'NCE OR OLD DRIFT RATE | DCS VAR'NCE OR DCS BAR OR ARB DRIFT FACTOR | NCB ADDRESS | RPH ADDRESS |

**Byte (hex)**
        **Contents**
00–02  Record ID: C"ARQ"
03  Trace instance

        **Code   Meaning**

        **C**        Competer (values before competer code executes).

        **D**        Drift detected.

        **N**        Entry (values on entry).

| | R | Drift reduction. |
|---|---|---|
| | X | Exit (values on exit). |
| **04** | | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | | 0 |
| **06–07** | | |

| Code | Meaning |
|---|---|
| N, X | Phase flags (ARB2_PHASE_FLAGS, ARB2_PHASE_FLAGS2). |
| C | Window size (ARB2_POT_WINDOW_SIZE). |
| D, R | 0 |

Phase flags:

| Bit | Meaning |
|---|---|
| 1... .... .... .... | Conservation phase |
| .1.. .... .... .... | Medium aggressive phase |
| ..1. .... .... .... | Send setup segment |
| ...1 .... .... .... | High flag |
| .... 1... .... .... | Delay change sum flag |
| .... .1.. .... .... | Upward cycle flag |
| .... ..1. .... .... | Idle - no data has been sent recently |
| .... ...1 .... .... | Startup |
| .... .... 1... .... | Last parity bit received on an ARB rate request |
| .... .... .1.. .... | Current ARB request parity bit |
| .... .... ..11 1111 | Not used |

| **08–0B** | | |

| Code | Meaning |
|---|---|
| C, N, X | Receiver threshold in microseconds (ARB2_POT_RCVR_THRESHOLD). |
| D | Current accumulated queuing time in microseconds (ARBLV_CLOCK_DRIFT). This is the minimum of the ACCUM QUEUING TIME and DCS BAR. |
| R | Current accumulated queuing time in microseconds (ARBLV_CLOCK_DRIFT). This is the absolute value of ACCUM QUEUING TIME plus 1500 microseconds. |

| **0C–0F** | | |

| Code | Meaning |
|---|---|

    **C, N, X**
        Maximum present threshold in microseconds
        (ARB2_POT_PRESENT_THR_MAX).

    **D**       Elapsed time, in microseconds, since a drift condition was detected.

    **R**       ACCUM QUEUING TIME has been rounded back to 0.

**10–13**

| Code | Meaning |
| --- | --- |
| **C** | Minimum present threshold in microseconds (ARB2_POT_PRESENT_THR_MIN). |
| **N, X** | Anticipated variance in milliseconds (ARB2_POT_ANTICIPATED_VAR). |
| **D, R** | Amount of clock drift, measured in microseconds, allowed per second. |

**14–17**

| Code | Meaning |
| --- | --- |
| **C** | Delay change sum variance. |
| **N, X** | Delay change sum bar in microseconds (ARB2_POT_DCS_BAR). |
| **D, R** | This value has a range of 2 to 20000. The default is 20000, which is associated with a drift of 50 microseconds per second. The drift rate, in microseconds, can be determined by dividing one million by the ARB_DRIFT_FACTOR. |

**18–1B**  NCB address
**1C–1F**  Request parameter header (RPH) address

## ARQ2 entry for RTP session data request (Part 2)

**Entry:** **ARQ2**
**VIT option:**
    HPR
**Event:** Any session data request is received over an RTP connection using
    Responsive Mode ARB.
**VIT processing module:**
    ISTITCHR

This trace record is a continuation of the ARQ entry. This trace record is written
only for the Entry and Exit instances.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 | 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D | E F |
|---|---|---|---|---|---|---|---|---|
| ARQ2 | CURRENT QUEUING TIME OR INCRE'T TREND | ACCUM QUEUING TIME | RECEIVR THRESH MAXIMUM OR RATE SCALER | RECEIVR THRESH MINIMUM OR ERROR COUNT | TARGET RATE | RECOVRY FACTOR | LAST REQUEST · CORRELATOR | 0 |

**Byte (hex)**

**Contents**

00–03   Record ID: C"ARQ2"

04–07   Current queuing time in milliseconds for Entry. Increment trend (ARB2_POT_INC_TREND) for Exit.

08–0B   Accumulated queuing time in microseconds (ARB_ACCUM_QTIME for Entry, ARB2_POT_ACCUM_QTIME for Exit)

0C–0F   Receiver threshold maximum in microseconds (ARB2_RCVR_THRESHOLD_MAX) for Entry. Rate scaler for Exit.

10–13   Receiver threshold minimum in microseconds (ARB2_RCVR_THRESHOLD_MIN) for Entry. Error count (ARB2_POT_ERROR_COUNT) for Exit.

14–17   Target rate (ARB2_TARGET_RATE)

18–1B   Recovery factor (ARB2_POT_RECOVERY_FACTOR )

1C–1D

     Last request correlator (ARB2_LAST_REQ_CORRELATOR)

1E–1F   0

# ASNB entry for ASSIGN_BUFFER requests

**Entry: ASNB**

**VIT option:**
     CSM

**Event:** IVTCSM REQUEST=ASSIGN_BUFFER

**VIT processing module:**
     ISTITCCS

**Control is returned to:**
     IVTSMCBF

This record provides the status of an IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction. A pair of ASNB records is created for each IVTCSM REQUEST=ASSIGN_BUFFER macroinstruction. The first record, along with the ASN2 continuation records, is created when the macroinstruction is issued and records the buffer token provided. The second record is created when the macroinstruction completes and records the new buffer tokens as well as additional information.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASNB | | | | I D | 0 | F L A G | R E C N U M | T Y P E | 0 | O W N E R | | RTN COD | | RSN COD | | UTILRTN CALLER OR RETURN ADDRESS | | | | LAST BUFFER LIST ENTRY | | | | NUMBER OF BUFFERS | | | | THREAD VALUE OR 0 | | | |

**Byte (hex)**

**Contents**

00–03   Record ID: C"ASNB"

04   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05   0

06   Trace record flags:

**B'0... ....'**
>   Indicates that this is the last record for this event.

**B'1... ....'**
>   Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**B'.1.. ....'**
>   Traces the issuance of the request.

**B'.0.. ....'**
>   Traces the completion of the request.

07   Trace record number to correlate all the entries for this particular event.

08   Type or state of buffers allocated

**X'80'**   Fixed

**X'20'**   Eligible to be made pageable

09   0

0A–0B   Owner ID. When traced on issuance, the owner ID is always 0.

0C–0D

>   Return Code

0E–0F   Reason Code

10–13   Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

14–17   Last buffer list entry. When traced on issuance, the last entry is nonzero whether an error occurred or not. On completion of assigned buffer, the last entry is 0 if the return code is 0 and nonzero if the return code is nonzero.

18–1B   Number of buffers requested to be assigned

1C–1F   THREAD value, if specified, or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

# ASN2 entry for ASSIGN_BUFFER requests

**Entry:  ASN2**

**VIT option:**
>   CSM

**Event:**  IVTCSM REQUEST=ASSIGN_BUFFER

**VIT processing module:**
ISTITCCS

This trace record is a continuation of the ASNB trace record. It contains information about the buffer images that were assigned. Note that two buffer tokens are present in the trace record for each buffer list entry. This is because a token is received on input and a new token is created to represent the new image of the buffer on output.

| 0 0 | 0 0 | 0 0 | 0 | 0 | 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 | 7 | 8 9 A B C D E F 0 1 2 3 | 4 5 6 7 8 9 A B C D E F |
| ASN2 | | 0 | F L A G | R E C N U M | BUFFER TOKEN FOR BUFFER LIST ENTRY | BUFFER TOKEN FOR BUFFER LIST ENTRY OR 0 |

**Byte (hex)**
**Contents**
00–03  Record ID: C"ASN2"
04–05  0
06  Trace record flag:

**B'0... ....'**
Indicates that this is the last record for this event.

**B'1... ....'**
Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.
07  Trace record number to correlate all the entries for this particular event
08–13  Buffer token contained in the buffer list entry
14–1F  Buffer token contained in the buffer list entry or 0 if there was an odd number of buffers in the list

# ATSK entry for attach of a subtask

**Entry:  ATSK**
**VIT option:**
PSS
**Event:  ATTACH of a subtask**
**VIT processing module:**
ISTRACPS
**Control is returned to:**
The module that issued the ATTACH

This trace record is written when a VTAM module issues an ATTACH macro to start another subtask. This record is generated after the ATTACH completes to ensure that completion information is included in the entry. Consequently, this entry can sometimes appear after the BTSK entry that is generated by the subtask it is attaching.

Not all subtask events generate this entry.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| ATSK | | I D | R C | 0 | TASK NAME | TCB ADDR | 0 | ISSUER ADDRESS | REG 1 |

**Byte (hex)**
>   **Contents**
**00–03** Record ID: C"ATSK"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Return code from the ATTACH macro
**06–07** 0
**08–0F** Name of the subtask being attached
**10–13** TCB address returned from the ATTACH macro
**14–17** 0
**18–1B** Address of the issuer of the ATTACH macro
**1C–1F** Register 1

# ATTx entry for attention

**Entry: ATT**
**VIT option:**
>   CIO
**Event:** Attention
**VIT processing module:**
>   ISTRACCI
**Control is returned to:**
>   ISTTSCCA or ISTTSCLA

ATTI is written when an attention interrupt occurs for a channel-attached communication controller or an SNA cluster controller.

ATTL is written when an attention interrupt occurs for a channel-attached non-SNA device (device defined in a local non-SNA major node).

ATTT is written when an attention interrupt occurs for a TCP/IP DLC connection.

ATTX is written when an attention interrupt occurs for a channel-to-channel attachment, including an IBM 3172 Interconnect Nways Controller.

For record types with suffix I, X, or T, the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ATTI ATTL ATTT ATTX | | | | ID | STATE | CCWOP | 0 | DEVICE | | | | NCB ADDRESS | | | | FLAG BYTES | | | | DLC OR 0 | CODE | SENSE | | CSW | | | | | | | |

**Byte (hex)**

**Contents**

00–03 Record ID:

C"ATTI" for ICNCB
C"ATTL" for LDNCB
C"ATTT" for RWNCB
C"ATTX" for XCNCB

04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05 Link station state (NCBLNKST) for ICNCB, LDNCB, and RWNCB; station state (XCNSSFSM) for XCNCB

06 CCW opcode or channel

07 0

08–0B Channel device name in EBCDIC (either a device address or device number)

0C–0F NCB address

10–13 Flag bytes (NCBFLAGS)

14 0 or for ATTT:
- For the CDLC DLC: C
- For the LCS DLC: L
- For the CLAW DLC: W
- For the CTC DLC: X
- For the HYPERchannel DLC: H

15 I/O completion code (IOSCOD in the IOSB)

16–17 Sense data (IOSSNS of the IOSB)

18–1F Channel status word from the IOSB

# BSPx entry for ADD, DELETE, and FIND macros

**Entry:** BSPA, BSPD, or BSPF

**VIT option:**
NRM

**Event:** BSBPCID ADD, DELETE, and FIND macros

**VIT processing module:**
ISTRACNR

**Control is returned to:**
ISTTSCPU

This trace record is written when ADD, DELETE, and FIND operations are performed for the BSBPCID tree. It shows:
- Key used for the invocation
- Information from the boundary session block (BSB) indicating the state of the session

Use this entry to track activity on a session.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| BSPA BSPF BSPD | I D | P L U S T A T E | M A C R O R C | F L A G S O R 0 | ADDRESS OF ISSUER OF MACRO | BSB ADDRESS | PCID | BSB FLAGS | RPH ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID:
   C"BSPA" for BSBPCID ADD
   C"BSPF" for BSBPCID FIND
   C"BSPD" for BSBPCID DELETE
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Primary logical unit (PLU) state
**06** Macro return code

   **Code**  **Meaning**
   **X'00'**  Successful invocation
   **X'04'**  Boundary session block (BSB) not found
   **X'08'**  Duplicate input address
   **X'12'**  Address not valid
**07** IP flags, if this BSB represents a SNA/IP session. 0 otherwise.
**08–0B** Address of the issuer of the macro
**0C–0F** Boundary session block (BSB) address
**10–17** Procedure-correlation identifier (PCID)
**18–1B** Boundary session block (BSB) flags (FLAG1, FLAG2, FLAG3)
**1C–1F** Request parameter header (RPH) address of module issuing the call

# BSSx entry for Add, Delete, and Find

**Entry:  BSSA, BSSD, or BSSF**
**VIT option:**
   NRM
**Event:  BSBSA ADD, DELETE, and FIND macros**
**VIT processing module:**
   ISTRACNR
**Control is returned to:**
   ISTTSCB2

This trace record contains information about BSBSA ADD, DELETE, or FIND macros issued by a VTAM module.

| 00 01 02 03 | 04 | 05 06 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 | 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C | 1D | 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BSSA BSSD BSSF | ID | 0 | RETURN ADDRESS | SUBAREA ADDRESS | SA ADDRESS ELEMENT | REX ADDRESS ELEMENT | BSB ADDRESS OR 0 | BSB FLAGS | PLU STATE | RETURN CODE | REX INDEX ELEMENT |

**Byte (hex)**

**Contents**

**00–03** Record ID:
- C"BSSA" for BSBSA ADD
- C"BSSD" for BSBSA DELETE
- C"BSSF" for BSBSA FIND

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** 0

**08–0B** Address of the issuer of the macro

**0C–0F** Subarea address

**10–11** Subarea element address

**12–13** Rex element address

**14–17** Boundary session block (BSB) address or 0

**18–1B** BSB Flags (FLAG1, FLAG2, FLAG3)

**1C** PLU state

**1D** Return code

> **Note:** It is not required that users of the product know the meaning of this internal VTAM return code. When required, the product support organization may use it to assist in internal flow diagnosis.

**1E–1F** Rex element address index

# BSXx entry for Add, Delete, and Find

**Entry: BSXA, BSXD, or BSXF**

**VIT option:**
NRM

**Event:** BSBREX ADD, DELETE, and FIND macros

**VIT processing module:**
ISTRACNR

**Control is returned to:**
ISTTSCB1

This trace record contains information about BSBREX ADD, DELETE, or FIND macros issued by a VTAM module.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BSXA BSXD BSXF ||||| ID | 0 ||| RETURN ADDRESS |||| LFSID || PUELE | ADDR | CRAFL | 0 ||| BSB ADDRESS OR 0 |||| FLAGS |||| STATE | RTNCD | 0 ||

**Byte (hex)**

**Contents**

**00–03**   Record ID:
- C"BSXA" for BSBREX ADD
- C"BSXD" for BSBREX DELETE
- C"BSXF" for BSBREX FIND

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07**   0

**08–0B**   Address of the issuer of the macro

**0C–0D**

   LFSID

**0E–0F**   PU element address

**10**   CRA flags

**11–13**   0

**14–17**   Boundary session block (BSB) address or 0

**18–1B**   BSB Flags (FLAG1, FLAG2, FLAG3)

**1C**   PLU state

**1D**   Return code

> **Note:** It is not required that users of the product know the meaning of this internal VTAM return code. When required, the product support organization may use it to assist in internal flow diagnosis.

**1E–1F**   0

# BTSK entry for begin a subtask

**Entry:  BTSK**
**VIT option:**
   PSS
**Event:**  Begin a subtask
**VIT processing module:**
   ISTRACPS
**Control is returned to:**
   The subtask that is just starting

This trace record is written when a VTAM subtask begins execution. This record can appear in the internal trace table before the ATSK entry for the ATTACH of the subtask.

Some subtasks start without generating this entry.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| BTSK | I D | 0 | TASK NAME | TCB ADDR | ENTRY POINT ADDR | 0 | REG 1 |

**Byte (hex)**

**Contents**

**00–03** Record ID: "BTSK"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** 0

**08–0F** Name of the subtask that was just entered

**10–13** TCB address of the current subtask

**14–17** Entry point into the subtask

**18–1B** 0

**1C–1F** Contents of register 1 upon entry

# BUFF entry for 8K external buffer

**Entry: BUFF**

**VIT option:**

None (generated when VIT is running externally using 8K buffers)

**Event:** External VIT tracing

**VIT processing module:**

ISTRACRR

**Control is returned to:**

ISTRACRR

This trace record is the first record in an 8K external VIT buffer. It determines if the 8K buffer was written to the external trace file in the proper sequence. It also indicates whether enough 8K buffers were specified on the TRACE start option or the MODIFY TRACE command by counting the records that were individually GTRACEd.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| BUFF | I D | 0 | BFR NUM | RECORDS INDIV GTRACED | BUFFER SEQ NUMBER | TIME STAMP OF FIRST RECORD IN BUFFER | 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: "BUFF"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06–07** Current BFRNUM value (as specified on TRACE start option or MODIFY TRACE command)

**08–0B** Count of VIT records individually GTRACEd due to a lack of 8K buffers

0C–0F   Buffer sequence number
10–17   Time stamp of first VIT record in the buffer
18–1B   0
1C–1F   RPH address

## CCI or CCO entry for SSCP (RUPE — Part 1)

**Entry:  CCI or CCO**
**VIT option:**
        SSCP
**Event:**  Requests/responses with a RUPE (Part 1)
**VIT processing module:**
        ISTRACSC
**Control is returned to:**
        ISTINCCI for inbound processing
        ISTINCCO for outbound processing

This trace record provides information about outbound processing done by
ISTINCCO and inbound processing done by ISTINCCI. ISTINCCO processes a
request; ISTINCCI sends a response to the requester.

If this entry is associated with an event failure (that is, the sense data is nonzero),
this entry is generated whether the SSCP option is in effect or not. It is treated as
an exception condition and, therefore, is traced whenever the VIT is active.

| 00 01 02 | 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 18 19 | 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| CCI OR CCO | FLAGS | ID | CBID | CPCBFL | CPCBRC | SAVE AREA ADDRESS | SAVE AREA ID | RETURN ADDRESS | ORIGIN NETWORK ADDRESS | DESTINATION NETWORK ADDRESS |

**Byte (hex)**
        **Contents**
00–02   Record ID:
        • C"CCI" for inbound processing
        • C"CCO" for outbound processing
03      Flags
        • 80 = response RU
        • 40 = sense traced
        • 00 = request RU
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is
        greater than X'FF'.
05      Control block ID=X'54'
06      Flags (CPCBFL)
07      Return code (CPCBRC)
08–0B   Save area address
0C–0F   Save area ID (bytes 4, 5, 7, and 8 of the module name if available)
10–13   Address of the issuer of the macro (CPCALL, CPEXIT, or CPWAIT)
14–19   For sender, network address at origin (RUPE)
1A–1F   For sender, network address at destination (RUPE)

## CC2 entry for SSCP (RUPE — Part 2)

**Entry:   CC2**
**VIT option:**
      SSCP
**Event:**  Requests/responses with a RUPE (Part 2)
**VIT processing module:**
      ISTRACSC

This trace record is a continuation of the CCI or CCO (RUPE) entry.

| 0 0 0 | 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| CC2 | C B I D | FIRST 24 BYTES OF RU | SENSE DATA OR 0 |

**Byte (hex)**
      **Contents**
**00–02**   Record ID: C"CC2"
**03**       Control block ID=X'54'
**04–1B**   First 24 bytes of RU
**1C–1F**   Sense data or 0

---

# CCI or CCO entry for SSCP (NCSPL — Part 1)

**Entry:   CCI or CCO**
**VIT option:**
      SSCP
**Event:**  Requests/responses for a NCSPL (Part 1)
**VIT processing module:**
      ISTRACSC
**Control is returned to:**
      ISTINCCI for inbound processing
      ISTINCCO for outbound processing

This trace record provides information about outbound processing, which processes a request, and inbound processing, which sends a response to the requester.

If this entry is associated with an event failure (that is, the sense data is nonzero), this entry is generated whether the SSCP option is in effect or not. It is treated as an exception condition and, therefore, is traced whenever the VIT is active.

| 0 0 | 0 | 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 | 1 | 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 | 9 | A B C D E F |
| CCI OR CCO | F L A G S | I D | C B I D | F L A G S | R T N C D | SAVE AREA ADDRESS | SAVE AREA ID | RETURN ADDRESS | CPCB OPCODE | T Y P E | 0 | DESTINATION NETWORK ADDRESS |

**Byte (hex)**

**Contents**

**00–02** Record ID:
- C"CCI" for inbound processing
- C"CCO" for outbound processing

**03** Flags
- 80 = response RU
- 40 = sense traced
- 00 = request RU

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID=X'60'

**06** Flags (CPCBFL)

**07** Return code (CPCBRC)

**08–0B** Save area address

**0C–0F** Save area ID (bytes 4, 5, 7, and 8 of the module name if available)

**10–13** Address of the issuer of the macro (CPCALL, CPEXIT, or CPWAIT)

**14–17** CPCB operation code (see *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)

**18** WTD type

**19** 0

**1A–1F** Network address at destination

# CC2 entry for SSCP (NCSPL — Part 2)

**Entry:** CC2

**VIT option:**
SSCP

**Event:** Requests/responses for a NCSPL (Part 2)

**VIT processing module:**
ISTRACSC

This trace record is a continuation of the CCI or CCO (NCSPL) entry.

| 0 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CC2 | C B I D | RDTE STATE | CPCB WTD | 0 |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"CC2"

03 Control block ID=X'60'
04–07 RDTE state (in control block RPRE)
08–0B CPCB WTD (first four bytes)
0C–1F 0

## CCI or CCO entry for SSCP (not RUPE or NCSPL)

**Entry:  CCI or CCO**
**VIT option:**
  SSCP
**Event:  Requests/responses for neither RUPE nor NCSPL**
**VIT processing module:**
  ISTRACSC
**Control is returned to:**
  ISTINCCI for inbound processing
  ISTINCCO for outbound processing

This trace record provides information about outbound processing, which processes a request, and inbound processing, which sends a response to the requester.

If this entry is associated with an event failure (that is, the sense data is nonzero), this entry is generated whether the SSCP option is in effect or not. It is treated as an exception condition and, therefore, is traced whenever the VIT is active.

| 00 01 02 | 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CCI OR CCO | FLAGS | ID | CBID | CPCBF | RTNCD | SAVE AREA ADDRESS | SAVE AREA ID | RETURN ADDRESS | CPCB OPCODE | CPCB WORD 3 | CPCB WORD 4 |

**Byte (hex)**
  **Contents**
00–02 Record ID:
  • C"CCI" for inbound processing
  • C"CCO" for outbound processing
03 Flags
  • 80 = response RU
  • 40 = sense traced
  • 00 = request RU
04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05 Control block ID (Control block IDs are shown in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)
06 CPCB flag (CPCBFL)
07 Return code (CPCBRC)
08–0B Save area address
0C–0F Save area ID (bytes 4, 5, 7, and 8 of the module name if available)
10–13 Address of the issuer of the macro (CPCALL, CPEXIT, or CPWAIT)
14–17 CPCB operation code (refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)
18–1B CPCB word 3 (field WTDPTR)

1C–1F  CPCB word 4 (field CPCBPH)

# CDHx entry for CIDCTL HOSTNODE Add, Change, Delete, and Find macros

**Entry:  CDHA, CDHC, CDHD, CDHF**
**VIT option:**
     NRM
**Event:** CIDCTL HOSTNODE ADD, CHANGE, DELETE, FIND macros
**VIT processing module:**
     ISTRACNR
**Control is returned to:**
     ISTNACRT

This trace record is written whenever a CIDCTL HOSTNODE ADD, CHANGE, DELETE, or FIND macro is invoked. CIDCTL FIND macro invocations executed during the process of sending or receiving data will not be traced unless they result in a nonzero return code.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 | 12 13 | 14 15 | 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CDHA CDHC CDHD CDHF | ID | 0 | FLAGS | RC | RETURN ADDRESS | CONTROL BLOCK ADDRESS | INDEX | SUB | ELEMENT | 0 | HOST NODE TABLE ENTRY ADDRESS | RPH ADDRESS |

**Byte (hex)**
     **Contents**
00–03  Record ID:
- C"CDHA" CIDCTL HOSTNODE ADD
- C"CDHC" CIDCTL HOSTNODE CHANGE
- C"CDHD" CIDCTL HOSTNODE DELETE
- C"CDHF" CIDCTL HOSTNODE FIND

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05  0
06  NATFLAGS byte from NATBL
07  Return code
08–0B  Return address of the invoker of the CIDCTL macro
0C–0F  Address of the control block that was added, found, or changed; 0 for delete
10–11  Index portion of subarea address
12–13  Subarea portion of subarea address
14–15  Element portion of subarea address
16–17  0
18–1B  Host node table entry address
1C–1F  Address of RPH

# CDNx entry for CIDCTL NEXTNODE Add, Delete, and Find macros
**Entry:  CDNA, CDND, CDNF**

**VIT option:**
>   NRM
**Event:**  CIDCTL NEXTNODE ADD, DELETE, FIND macros
**VIT processing module:**
>   ISTRACNR
**Control is returned to:**
>   ISTNACRT

This trace record is written whenever a CIDCTL NEXTNODE ADD, DELETE, or FIND macro is invoked. CIDCTL FIND macro invocations executed during the process of sending or receiving data will not be traced unless they result in a nonzero return code.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 | 12 13 | 14 15 | 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CDNA CDND CDNF | ID | 0 | FLAGS | RC | RETURN ADDRESS | CONTROL BLOCK ADDRESS | INDEX | SUB | ELEMENT | 0 | HOST NODE TABLE ENTRY ADDRESS | RPH ADDRESS |

**Byte (hex)**
>   **Contents**
**00–03**  Record ID:
>   - C"CDNA" CIDCTL NEXTNODE ADD
>   - C"CDND" CIDCTL NEXTNODE DELETE
>   - C"CDNF" CIDCTL NEXTNODE FIND

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  0
**06**  NATFLAGS byte from NATBL
**07**  Return code
**08–0B**  Return address of the invoker of the CIDCTL macro
**0C–0F**  Address of the control block that was added or found; 0 for delete
**10–11**  Index portion of subarea address
**12–13**  Subarea portion of subarea address
**14–15**  Element portion of subarea address
**16–17**  0
**18–1B**  Host node table entry address
**1C–1F**  Address of RPH

# CDN2 entry for CIDCTL NEXTNODE FIND (Part 2)

**Entry:**  CDN2
**VIT option:**
>   NRM
**Event:**  CIDCTL NEXTNODE FIND macro
**VIT processing module:**
>   ISTRACNR

This trace record is a continuation of the CDNF entry. This entry is present only when the subordinate host node table entry address is found and is associated

with a boundary logical unit control block (BLB).

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CDN2 | SUB-HNTE ADDRESS | BLB ADDR OR 0 | ZEROS |

**Byte (hex)**

**Contents**

00–03  Record ID: C'CDN2'

04-07  Subordinate host node table entry address.

08-0B  Boundary Logical Unit control block address or 0.

0C-0F  0

# CFAL entry for coupling facility IXLALTER service (Part 1)

**Entry:  CFAL**

**VIT option:**

CFS

**Event:**  Invocation of the MVS macro IXLALTER

**VIT processing module:**

ISTRACCF

**Control is returned to:**

ISTFSUAL

This trace entry is written when the IXLALTER macro is used to:
- Initiate an alter of the coupling facility storage attributes
- Stop an alter in progress of the coupling facility storage

| 0 0 0 0 | 0 4 | 0 5 | 0 6 | 0 7 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 0 4 | 0 5 | 0 6 | 0 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CFAL | I D | 0 | F U N C T I O N | F L A G S | CFSSTR ADDRESS | 0 | RETURN ADDRESS | MVS RETURN CODE | MVS REASON CODE | RPH ADDRESS |

**Byte (hex)**

**Contents**

00–03  Record ID: C"CFAL"

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05  0

06  Function request by this invocation of IXLALTER:

**Bit  Meaning**

**X'01'**  REQUEST(START) to start the alter process.

**X'02'**  REQUEST(STOP) stop the alter process.

**07**     Event flags

> **Bit     Meaning**
> **1... ....**
> > Request that the size of the coupling facility structure be changed.
>
> **.1.. ....**
> > Request that the entry-to-element ratio for a coupling facility structure be changed.
>
> **..xx xxxx**
> > 0

**08–0B**  Structure object (CFSSTR) address.
**0C–0F**  0
**10–13**  Return address of the invoker of the utility module.
**14–17**  Return code from the MVS macro IXLALTER. Refer to *z/OS MVS Programming: Sysplex Services Reference* for an explanation of this code.
**18–1B**  Reason code from the MVS macro IXLALTER. Refer to *z/OS MVS Programming: Sysplex Services Reference* for an explanation of this code.
**1C–1F**  Request parameter header (RPH) address.

# CFA2 entry for coupling facility IXALTER service (Part 2)

**Entry:  CFA2**
**VIT option:**
> CFS

**Event:** Invocation of the MVS macro IXLALTER
**VIT processing module:**
> ISTRACCF

This trace entry is a continuation of the CFAL entry.

| 00 01 02 03 | 04 05 | 06 07 | 08 09 0A 0B | 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|
| CFA2 | TARGET RATIO | TARGET ELEMENT RATIO | TARGET STRUCTURE SIZE | 0 |

**Byte (hex)**
> **Contents**

**00–03**  Record ID: C"CFA2"
**04–05**  If request is to alter the entry-to-element ratio, this is the target entry portion of the ratio.
**06–07**  If request is to alter the entry-to-element ratio, this is the target element portion of the ratio.
**08–0B**  If request is to alter the structure size, this is the target structure size.
**0C–1F**  0

# CFCE entry for coupling facility request completion exit

**Entry:** CFCE
**VIT option:**
   CFS
**Event:** Complete exit driven for a connection to coupling facility structure
**VIT processing module:**
   ISTRACCF
**Control is returned to:**
   ISTFSXCM

This trace record is written when a previous request to access the coupling facility structure has completed asynchronously.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| CFCE | ID | FLAGS | 0 | CONNECT ID | CFSSTR ADDRESS | REQUEST DATA | MVS RETURN CODE | MVS REASON CODE | RPH ADDRESS |

**Byte (hex)**
   **Contents**
**00–03** Record ID: C"CFCE"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Event flags:

   **Bit** **Meaning**
   **1... ....**
      This event applies to the new version of the structure.
   **.1.. ....**
      This event pertains to an invocation of the IXLLIST macro.
   **..1. ....**
      This event pertains to an invocation of the IXLCACHE macro.
   **...1 ....**
      This event pertains to an invocation of the IXLLOCK macro.
   **.... xxxx**
      0

**06** 0
**07** The connection identifier for the target of this event
**08–0B** Structure object (CFSSTR) address
**0C–13** The request data specified on the MVS macro invocation identified in the flags at offset X'05' of CFCE

   **Bit** **Meaning**
   **..1. ....**
      When this bit is on at the completion exit, the buffer contents for the DELETE_ENTRYLIST are traced using the CFLS VIT entries.
**14–17** Return code for the MVS macro indicated at offset X'05'
**18–1B** Reason code for the MVS macro indicated at offset X'05'

1C–1F   Request parameter header (RPH) address

# CFCN entry for coupling facility connection service (Part 1)

**Entry:  CFCN**
**VIT option:**
        CFS
**Event:**  Invocation of the MVS macro IXLCONN
**VIT processing module:**
        ISTRACCF
**Control is returned to:**
        ISTFSGCN

This trace record is written when there is an attempted connection to the coupling facility structure with the IXLCONN macro.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFCN | | | | ID | FLAGS | 0 | ID OR 0 | CFSSTR ADDRESS | | | | 0 | | | | INVOKER ADDRESS | | | | MVS RETURN CODE | | | | MVS REASON CODE | | | | RPH ADDRESS | | | |

**Byte (hex)**
        **Contents**
**00–03**   Record ID: C"CFCN"
**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**      Event flags:

        **Bit      Meaning**
        **1... ....**
                The rebuild process is in progress for the structure.
        **.1.. ....**
                The rebuild stop process is in progress for the structure.
        **..1. ....**
                This connection attempt caused the structure to be allocated.
        **...1 ....**
                A previously failed connection has been reestablished.
        **.... 1...**
                Full connectivity to the rebuild structure cannot be obtained. This bit is valid only if this connection attempt caused the structure to be allocated. This bit will be set only if IXLCONN was invoked during rebuild processing.
        **.... .1..**
                The structure involved is volatile.
        **.... ..1.**
                VTAM did not process this event.
        **.... ...x**
                0
**06**      0

| | |
|---|---|
| **07** | Connection identifier or 0. If IXLCONN was successful, this field will contain the connection identifier assigned to this connection. Otherwise this field will contain the value 0. |
| **08–0B** | Structure object (CFSSTR) address |
| **0C–0F** | 0 |
| **10–13** | The address of the invoker of IXLCONN |
| **14–17** | Return code for the MVS macro IXLCONN |
| **18–1B** | Reason code for the MVS macro IXLCONN |
| **1C–1F** | Request parameter header (RPH) address |

# CFC2 entry for coupling facility connection service (Part 2)

**Entry:  CFC2**
**VIT option:**
    CFS
**Event:**  Invocation of the MVS macro IXLCONN
**VIT processing module:**
    ISTRACCF

This trace record is a continuation of the CFCN entry.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 | 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D | E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CFC2 | ACTUAL SIZE OR 0 | MAX SIZE | ENTRYO ENATTRIO | ELEMO ERATMIO | 0 |

**Byte (hex)**
    **Contents**

| | |
|---|---|
| **00–03** | Record ID: C"CFC2" |
| **04–07** | The actual size of the structure allocated, or 0. If IXLCONN was successful, this field contains the actual number of 4K blocks allocated to the structure. |
| **08–0B** | Maximum structure size. This field contains the structure size, in 4K blocks, defined in the active coupling facility resource management policy at the time this structure was allocated. |
| **0C–0D** | The entry portion of the entry-to-element ratio specified on IXLCONN. |
| **0E–0F** | The element portion of the entry-to-element ratio specified on IXLCONN. |
| **10–1F** | 0 |

# CFDS entry for coupling facility disconnection service (Part 1)

**Entry:  CFDS**
**VIT option:**
    CFS
**Event:**  Invocation of the MVS macro IXLDISC
**VIT processing module:**
    ISTRACCF
**Control is returned to:**
    ISTFSUDC

This trace record is written when a connector disconnects from the coupling facility structure with the IXLDISC macro.

| 0 0 | 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|
| CFDS | ID | FLAGS | 0 | CONNECT ID | CFSSTR ADDRESS | 0 | RETURN ADDRESS | RETURN CODE | REASON CODE | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFDS"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Macro parameter flags

**Bit**     **Meaning**

**1... ....**
REASON(FAILURE) was specified for the request

**.xxx xxxx**
0

**06** 0

**07** Connection identifier of the invoker of IXLDISC

**08–0B** Structure object (CFSSTR) address

**0C–0F** 0

**10–13** The address of the invoker of IXLDISC

**14–17** Return code for the MVS macro IXLDISC

**18–1B** Reason code for the MVS macro IXLDISC

**1C–1F** Request parameter header (RPH) address

# CFD2 entry for coupling facility disconnection service (Part 2)

**Entry: CFD2**

**VIT option:**
CFS

**Event:** Invocation of the MVS macro IXLDISC

**VIT processing module:**
ISTRACCF

This trace record is a continuation of the CFDS entry.

| 0 0 0 0 0 1 2 3 | 0 0 0 0 4 5 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|---|---|
| CFD2 | CONNECTOR SPECIFIC DISCONNECT DATA | MAX SIZE | 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFD2"

**04–07** Connector specific disconnect data

**08–0B** Maximum size

**0C–1F** 0

# CFEE entry for coupling facility connection event exit

**Entry: CFEE**

**VIT option:**

CFS

**Event:** Event Exit Driven for a Connection to a Coupling Facility Structure

**VIT processing module:**

ISTRACCF

**Control is returned to:**

ISTFSXEV

This trace record is written when any of the following actions occurs:
- A new connection is established to the coupling facility structure.
- A connection to the coupling facility structure is terminated.
- A connector loses read/write access to the coupling facility structure.
- A state change occurs for the coupling facility structure.
- A user sync point is set by a connector to the coupling facility structure.
- A user sync point has been confirmed by all connectors to the coupling facility structure.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFEE | | | | I D | 0 | E V E N T | F L A G S | CFSSTR ADDRESS | | | | EVENT SEQ NUMBER | | | | S U B J E C T | S T A T E | S T A R T R S N N N | S T O P R S N | USER DATA 1 | | | | USER DATA 2 | | | | RPH ADDRESS | | | |

**Byte (hex)**

> **Contents**

**00–03**  Record ID: C"CFEE"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  0

**06**  Event identifier (refer to the mapping of IXLYEEPL in *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of this code.)

**07**  The event identifier in byte 06 determines the meaning of this field.

Event flags:

- If byte 06 indicates a loss of connectivity event, use the following flags:

  > **Bit      Meaning**
  >
  > **1... ....**
  > > The rebuild process is in progress for the structure.
  >
  > **.1.. ....**
  > > The rebuild stop process is in progress for the structure.
  >
  > **..1. ....**
  > > This event applies to the rebuild version of the structure.
  >
  > **...1 ....**
  > > No action will be taken. MVS will determine action based on policy information.
  >
  > **.... xxxx**
  > > 0

- If byte 06 indicates an existing connection event, use the following flags:

  > **Bit      Meaning**
  >
  > **1... ....**
  > > The rebuild process is in progress for the structure.
  >
  > **.1.. ....**
  > > The rebuild stop process is in progress for the structure.
  >
  > **..1. ....**
  > > This event does not pertain to a real connection but indicates that all existing connection events have been received.
  >
  > **...1 ....**
  > > The connection identified is active.
  >
  > **.... 1...**
  > > The connection identified does not have access to the structure.
  >
  > **.... .xxx**
  > > 0

- If byte 06 indicates a disconnected or failed event, use the following flags:

**Bit     Meaning**
**1... ....**
>   The rebuild process is in progress for the structure.

**.1.. ....**
>   The rebuild stop process is in progress for the structure.

**..1. ....**
>   This event applies to the rebuild version of the structure.

**...1 ....**
>   The connection disconnected abnormally.

**.... xxxx**
>   0

- If byte 06 indicates a rebuild existing connection event, use the following flags:

**Bit     Meaning**
**1... ....**
>   The rebuild process is in progress for the structure.

**.1.. ....**
>   The rebuild stop process is in progress for the structure.

**..1. ....**
>   This event does not pertain to a real connection but indicates that all rebuild existing connection events have been received.

**...1 ....**
>   The connection identified is active.

**.... xxxx**
>   0

- If byte 06 indicates structure alter begin event, use the following flags:

**Bit     Meaning**
**1... ....**
>   The rebuild process is in progress for the structure.

**.1.. ....**
>   The rebuild stop process is in progress for the structure.

**..1. ....**
>   Structure size will be altered.

**...1 ....**
>   Entry-to-element ratio will be altered.

**.... xxxx**
>   0

- If byte 06 indicates structure alter end event, use the following flags:

**Bit     Meaning**
**1... ....**
>   The rebuild process is in progress for the structure.

**.1.. ....**
>   The rebuild stop process is in progress for the structure.

**..1. ....**
>   Attempt to alter structure size.

**...1 ....**
>   Attempt to alter entry-to-element ratio.

**.... 1...**
>   Alter request able to meet all specified targets.

**.... .1..**
>   Alter request able to only meet some specified targets.

**.... ..xx**
>   0

- If byte 06 indicates a recommended action, use the following flags:

  **Bit**    **Meaning**
  **1... ....**
       The rebuild process is in progress for the structure.
  **.1.. ....**
       The rebuild stop process is in progress for the structure.
  **..1. ....**
       Policy is available to determine action.
  **...1 ....**
       Action is disconnect.
  **.... xxxx**
       0

- For all other events, use the following flags:

  **Bit**    **Meaning**
  **1... ....**
       The rebuild process is in progress for the structure.
  **.1.. ....**
       The rebuild stop process is in progress for the structure.
  **..1. ....**
       VTAM ignored this event.
  **...x xxxx**
       0

**08–0B**    Structure object (CFSSTR) address
**0C–0F**    Event sequence number. Each event is assigned a unique sequence number.
**10**    The connection identifier for the subject of this event
**11**    Connection FSM State. Connection states are listed in the VTAM data map ISTCFCON.
**12**    Reason for starting rebuild (refer to the mapping of IXLYEEPL in *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of this code).
**13**    Reason for stopping rebuild (refer to the mapping of IXLYEEPL in *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of this code).

- If byte 06 indicates a rebuild event, then this field contains the reason for stopping rebuild (refer to the mapping of IXLYEEPL in *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of this code).

- If byte 06 indicates structure alter end event, use the following flags:

  **Bit**    **Meaning**
  **1... ....**
       Alter request failed due to structure failure.
  **.1.. ....**
       Alter request failed due to loss of connectivity.
  **..1. ....**
       Alter request failed due to rebuild started.
  **...1 ....**
       Alter request failed because ratios specified on alter request are not consistent with structure attributes.
  **.... xxxx**
       0

**14–17**    User Data 1

     The contents of this field depend upon the event received and the state of the structure when the event was received.

     If this event pertains to a user sync point, this field contains the connector-defined event for the user sync point set if a user sync point has been set.

If this event pertains to a disconnect or failed connection event, then this field contains the first four bytes of the connector-defined data specified in IXLDISC.

If this event pertains to a rebuild connects complete event, then this field contains the number of active connectors to the original structure.

If this event pertains to a rebuild event, other than rebuild connects complete, then this field contains the connector-defined reason for starting rebuild, if a connector-specified reason was given.

If this event pertains to alter begin event, then the first two bytes contain the target entry portion of the entry-to-element ratio. The second two bytes contain the target element portion of the entry-to-element ratio.

If this event pertains to an alter end event and the ratio was changed, then this field contains the current number of entries.

If this event indicates a structure state change, the first two bytes of this field contain a validity flag. The validity flag byte 14 will contain:

| Bit | Meaning |
| --- | --- |
| 1... .... | Coupling facility operational level of the coupling facility in which the structure resides is valid. If this bit has been set, User Data 2 contains the coupling facility operational level for the coupling facility in which the structure resides. |

Otherwise, this field will contain all zeros.

**18–1B**  User Data 2

The contents of this field depend upon the event received and the state of the structure when the event was received.

If this event pertains to a user sync point, this field contains the connector-defined event for the user sync point confirmed if a user sync point has been confirmed.

If this event pertains to a disconnect or failed connection event, then this field will contain the last four bytes of the connector-defined data specified in IXLDISC.

If this event pertains to a rebuild stop event, then this field will contain the connector-defined reason for stopping the rebuild, if a connector-specified reason was given.

If this event pertains to a rebuild connects complete event, then this field contains the number of connectors that successfully connected to the rebuild structure.

If this event pertains to an alter begin event, then this field contains the target structure size.

If this event pertains to an alter end event and the ratio was changed, then this field contains the current number of elements. If the structure size was changed then this field contains the current structure size. If both ratio and size are changed, this field will contain the current number of elements.

If this event indicates a structure state change and byte 14 (first byte of User Data 1 field) contains a value of X'80', this field contains the coupling facility operational level. This is the operational level of the coupling facility in which the structure resides.

Otherwise, this field will contain all zeros.

1C–1F   Request parameter header (RPH) address

## CFER entry for coupling facility connection event exit response service

**Entry:** **CFER**
**VIT option:**
   CFS
**Event:** Invocation of the MVS macro IXLEERSP
**VIT processing module:**
   ISTRACCF
**Control is returned to:**
   ISTFSURS

This trace record is written when a response is given to an event exit event using the IXLEERSP macro.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| CFER | I D | S U B J E C T | E V E N T | C O N N E C T I D | CFSSTR ADDRESS | EVENT SEQ NUMBER | RETURN ADDRESS | RETURN CODE | REASON CODE | RPH ADDRESS |

**Byte (hex)**
**Contents**
**00–03**   Record ID: C"CFER"
**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**   The connection identifier of the subject of the original event. This field applies to Disconnected/Failed Events, Existing Connection Events, and Rebuild Connect Failure Events only.
**06**   Event code (Refer to the mapping of ISTXEEPL in *z/OS Communications Server: SNA Data Areas Volume 1* for an explanation of this code.)
**07**   Connection identifier of the invoker of IXLEERSP
**08–0B**   Structure object (CFSSTR) address
**0C–0F**   The event sequence number of the event to which a response is being given. This field applies to Disconnected/Failed Events, Existing Connection Events, and Rebuild Connect Failure Events only. Otherwise, this field contains a 0.
**10–13**   The address of the invoker of IXLEERSP
**14–17**   Return code for the MVS macro IXLEERSP
**18–1B**   Reason code for the MVS macro IXLEERSP
**1C–1F**   Request parameter header (RPH) address

## CFFC entry for coupling facility IXLFORCE service

**Entry:** **CFFC**
**VIT option:**
   CFS

**Event:** Invocation of the MVS macro IXLFORCE
**VIT processing module:**
 ISTRACCF
**Control is returned to:**
 ISTFSUFC

This trace record is written when a failed-persistent connection is deleted using the IXLFORCE macro.

| 0 0 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|
| CFFC | I D | S U B J E C T | F U N C | C O N N E C T  I D  CFSSTR ADDRESS | 0 | RETURN ADDRESS | RETURN CODE | REASON CODE | RPH ADDRESS |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"CFFC"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** The connection identifier of the failed-persistent connection
**06** The function requested by this invocation of IXLFORCE:
 **X'01'** REQUEST(CONNECTION) has been specified.
**07** Connection identifier of the invoker of IXLFORCE
**08–0B** Structure object (CFSSTR) address
**0C–0F** 0
**10–13** The address of the invoker of IXLFORCE
**14–17** Return code for the MVS macro IXLFORCE
**18–1B** Reason code for the MVS macro IXLFORCE
**1C–1F** Request parameter header (RPH) address

# CFLS entry for coupling facility IXLLIST service (Part 1)

**Entry: CFLS**
**VIT option:**
 CFS
**Event:** Invocation of the MVS macro IXLLIST
**VIT processing module:**
 ISTRACCF
**Control is returned to:**
 Module invoking the IXLLIST macro that caused the record to be produced.

This trace record is written for a request to access the coupling facility structure using the IXLLIST macro.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CFLS | I D | 0 | F U N C T I O N | F L A G S | CFSSTR ADDRESS | CFSBUF ADDRESS OR 0 | RETURN ADDRESS | RETURN CODE OR PART 1 OF REQUEST DATA | REASON CODE OR PART 2 OF REQUEST DATA | RPH ADDRESS |

**Byte (hex)**
    **Contents**

**00–03** Record ID: C"CFLS"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** The function requested by this invocation of IXLLIST:

    **X'01'** REQUEST(READ) specified
    **X'02'** REQUEST(READ_LCONTROLS) specified
    **X'03'** REQUEST(WRITE) specified
    **X'04'** REQUEST(MOVE) specified
    **X'05'** REQUEST(DELETE) specified
    **X'06'** REQUEST(LOCK) specified
    **X'07'** REQUEST(MONITOR_LIST) specified
    **X'08'** REQUEST(READ_LIST) specified
    **X'09'** REQUEST(DELETE_MULTI) specified
    **X'0A'** REQUEST(WRITE_LCONTROLS) specified
    **X'0B'** REQUEST(DELETE_ENTRYLIST) specified
    **X'0C'** REQUEST(READ_MULT) specified

**07** Flags

    **Bit**    **Meaning**
    **1... ....**
        Request will complete asynchronously. Bytes X'14' – X'1B' will contain the request data.
    **.1.. ....**
        Request is for the rebuild version of the structure.
    **..1. ....**
        The buffer list contains real addresses.
    **...x xxxx**
        0

**08–0B** Structure object (CFSSTR) address

**0C–0F** Buffer object (CFSBUF) address or 0 if not using a buffer object.

**10–13** The address of the invoker of IXLLIST

**14–17** Return code for the MVS macro IXLLIST or the first four bytes of request data. If this IXLLIST invocation will complete asynchronously, this field holds the first four bytes of the request data specified on this request. Otherwise it holds the return code for this invocation of IXLLIST.

**18–1B** Reason code for the MVS macro IXLLIST or the second four bytes of request data. If this IXLLIST invocation will complete asynchronously, this field holds the second four bytes of the request data specified on this request. Otherwise it holds the reason code for this invocation of IXLLIST.

When this field represents the second four bytes of the request data, the tracing of the buffer contents is deferred until the completion exit is driven, as follows:

**Bit      Meaning**
**..1.  ....**
When this bit is on at the completion exit, the buffer contents for the DELETE_ENTRYLIST are traced using the CFLS VIT entries.

**1C–1F**   Request parameter header (RPH) address

# CFL2 entry for IXLLIST lock service (Part 2)

**Entry: CFL2**
**VIT option:**
CFS
**Event:** Invocation of the MVS macro IXLLIST
**VIT processing module:**
ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a LOCK request.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFL2 | | | | F U N C T I O N | M O D E | L O C K O P E R | L O C K C O M P | LOCK INDEX | | | | 0 | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
**Contents**
**00–03**   Record ID: C"CFL2"
**04**   Function code = X'06'
**05**   The value specified for the MODE parameter:

- X'01'MODE(SYNCSUSPEND) specified
- X'02'MODE(SYNCEXIT) specified
- X'03'MODE(AYSNCEXIT) specified

**06**   The value specified for the LOCKOPER parameter:

- X'01'LOCKOPER(SET) specified
- X'02'LOCKOPER(RESET) specified

**07**   The value specified for the LOCKCOMP parameter. This 1-byte value represents the connection identifier of the connector that is assumed to own a given lock.
**08–0B**   The index of the lock that is being manipulated.
**0C–1F**   0

# CFL2 entry for IXLLIST list controls services (Part 2)

**Entry: CFL2**
**VIT option:**
CFS

**Event:** Invocation of the MVS macro IXLLIST with REQUEST
READ_LCONTROLS or WRITE_LCONTROLS
**VIT processing module:**
ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS
entry indicates a READ_LCONTROLS or WRITE_LCONTROLS request.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|
| CFL2 | F U N C T I O N | M O D E | 0 | SOURCE LIST NUMBER | NUMBER ENTRIES | LISTKEY | 0 | FIRST FOUR BYTES LIST DESC |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFL2"

**04** Function code:

- X'02'REQUEST(READ_LCONTROLS) specified

- X'0A'REQUEST(WRITE_LCONTROLS) specified

**05** The value specified for the MODE parameter:

- X'01'MODE(SYNCSUSPEND) specified

- X'02'MODE(SYNCEXIT) specified

- X'03'MODE(AYSNCEXIT) specified

**06–07** 0

**08–0B** The list number for which the data is being retrieved or updated

**0C–0F** If request is READ_LCONTROLS, the number of entries currently in use
on the list; otherwise 0

**10–13** If the request is WRITE_LCONTROLS, the LISTKEY value; otherwise 0

**14–1B** 0

**1C–1F** The first four bytes of the 32-byte list descriptor area

# CFL3 entry for IXLLIST service (REQUEST LCONTROLS) (Part 3)

**Entry: CFL3**
**VIT option:**
CFS
**Event:** Invocation of the MVS macro IXLLIST with REQUEST
READ_LCONTROLS or WRITE_LCONTROLS
**VIT processing module:**
ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS
entry indicates a READ_LCONTROLS or WRITE_LCONTROLS request.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| CFL3 | | | | LAST 28 BYTES OF LIST DESCRIPTOR | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
**Contents**
00–03   Record ID: C"CFL3"
04–1F   The last 28 bytes of the 32-byte list descriptor area

# CFL2 entry for IXLLIST services (single entry) (Part 2)

**Entry: CFL2**
**VIT option:**
    CFS
**Event:** Invocation of the MVS macro IXLLIST with REQUEST READ, WRITE, MOVE, or DELETE
**VIT processing module:**
    ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a READ, WRITE, MOVE, or DELETE request.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| CFL2 | | | | FUNCTION | MODE | DATAOPER | 0 | SOURCE LIST NUMBER | | | | BUFFER ALET OR LISTKEY INCREM | | | | ENTRYNAME, ENTRYID, OR ENTRYKEY | | | | | | | | | | | | | | | |

**Byte (hex)**
**Contents**
00–03   Record ID: C"CFL2"
04   Function code:
- X'01' REQUEST(READ) specified
- X'03' REQUEST(WRITE) specified
- X'04' REQUEST(MOVE) specified
- X'05' REQUEST(DELETE) specified

05   The value specified for the MODE parameter:
- X'01' MODE(SYNCSUSPEND) specified
- X'02' MODE(SYNCEXIT) specified
- X'03' MODE(AYSNCEXIT) specified

06   If byte X'04' indicates a MOVE or DELETE function, use the following to determine the DATAOPER parameter:
- X'01' DATAOPER(NONE) specified

　　• X'02'DATAOPER(READ) specified

　　• X'03'DATAOPER(WRITE) specified

**07**　　　0

**08–0B**　The list number for which the data is being retrieved or updated

**0C–0F**

　　• If the request is READ, WRITE, or DELETE, the Access List Entry Token (ALET) of the data buffers in the buffer object

　　• If the request is MOVE, the LISTKEY INCREMENT value

**10–1F**　The value of any ENTRYNAME, ENTRYID, or ENTRYKEY parameter specified

# CFL2 entry for IXLLIST services (multiple entries) (Part 2)

**Entry:　CFL2**

**VIT option:**

　　CFS

**Event:**　Invocation of the MVS macro IXLLIST with REQUEST READMULT, READLIST, or DELETEMULT

**VIT processing module:**

　　ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a READMULT, READLIST or DELETEMULT request.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CFL2 | | F U N C T I O N | M O D E | N U M B E R | P R O C E S S | SOURCE LIST NUMBER | BUFFER ALET | RESTART ENTRYID OR RESTART TOKEN |

**Byte (hex)**

**Contents**

**00–03**　Record ID: C"CFL2"

**04**　　　Function code:

　　• X'08'REQUEST(READ_LIST) specified

　　• X'09'REQUEST(DELETE_MULT) specified

　　• X'0C'READ(READ_MULT) specified

**05**　　　The value specified for the MODE parameter:

　　• X'01'MODE(SYNCSUSPEND) specified

　　• X'02'MODE(SYNCEXIT) specified

　　• X'03'MODE(AYSNCEXIT) specified

**06–07**　The number of processed entries. If the number is greater than 65635, the number will be X'FFFF'.

**08–0B**　The list number for which the data is being retrieved or deleted

**0C–0F**　Access List Entry Token (ALET) of the data buffers in the buffer object

**10–1F** For READLIST request, the 12-byte restart entry ID padded to the right with zeros. For READMULT and DELETEMULT, the 16-byte restart token.

# CFL3 entry for IXLLIST services (REQUEST entries) (Part 3)

**Entry: CFL3**
**VIT option:**
    CFS
**Event:** Invocation of the MVS macro IXLLIST with REQUEST READ, WRITE, MOVE or DELETE, READMULT, READLIST, or DELETEMULT
**VIT processing module:**
    ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a REQUEST READ, WRITE, MOVE or DELETE, READMULT, READLIST, or DELETEMULT request. The VTAM internal trace will generate up to five CFL3 entries.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CFL3 | 28 BYTES OF BUFFER LIST |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"CFL3"
**04–1F** Up to 28 bytes of the contents of the buffer list

# CFL2 entry for IXLLIST delete entry list service (Part 2)

**Entry: CFL2**
**VIT option:**
    CFS
**Event:** Invocation of the MVS macro IXLLIST with REQUEST DELETE_ENTRYLIST
**VIT processing module:**
    ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a DELETE_ENTRYLIST request.

| 0 0 0 0 | 0 4 | 0 5 | 0 0 6 7 | 0 0 8 9 | 0 0 A B | 0 0 0 0 C D E F | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|---|---|---|---|---|---|
| CFL2 | F U N C T I O N | M O D E | 0 | F I R S T | L A S T | BUFFER ALET | 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFL2"

**04** Function code = X'0B'

**05** The value specified for the MODE parameter:

- X'01'MODE(SYNCSUSPEND) specified

- X'02'MODE(SYNCEXIT) specified

- X'03'MODE(AYSNCEXIT) specified

**06–07** 0

**08–09** Index of the first entry ID processed

**0A–0B** Index of the last entry ID processed

**0C–0F** Access List Entry Token (ALET) of the buffer holding the entry IDs to be deleted

**10–1F** 0

## CFL3 entry for IXLLIST delete entry list service (Part 3)

**Entry: CFL3**

**VIT option:**

CFS

**Event:** Invocation of the MVS macro IXLLIST with REQUEST DELETE_ENTRYLIST

**VIT processing module:**

ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a REQUEST DELETE_ENTRYLIST request. The VTAM internal trace will generate up to 36 CFL3 entries.

| 0 0 0 0 0 1 2 3 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|
| CFL3 | 28 BYTES OF ENTRY ID BUFFER |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFL3"

**04–1F** Up to 28 bytes of the contents of the buffer containing entry IDs to be

deleted. The buffer contents traced will begin at the index of the first entry (specified at offset X'04' in CFL2) and end at the index of the second entry (specified at offset X'0C' in CFL2) if 36 CFL3 entries have not been generated.

## CFL2 entry for IXLLIST monitor service (Part 2)

**Entry: CFL2**
**VIT option:**
    CFS
**Event:** Invocation of the MVS macro IXLLIST with REQUEST MONITOR_LIST
**VIT processing module:**
    ISTRACCF

This trace record is a continuation of the CFLS entry when byte X'06' in the CFLS entry indicates a MONITOR_LIST request.

| 0 0 0 0 | 0 4 | 0 5 | 0 6 | 0 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|---|---|---|---|---|---|
| CFL2 | F U N C T I O N | M O D E | A C T I O N | 0 | SOURCE LIST NUMBER | VECTOR INDEX | 0 |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"CFL2"
**04**  Function code = X'07'
**05**  The value specified for the MODE parameter:
- X'01'MODE(SYNCSUSPEND) specified
- X'02'MODE(SYNCEXIT) specified
- X'03'MODE(AYSNCEXIT) specified

**06**  The value specified for the ACTION parameter:
- X'01'ACTION(START) specified
- X'02'ACTION(STOP) specified

**07**  0
**08–0B**  The list number for which the monitoring is being started or stopped
**0C–0F**  The list monitoring vector index being assigned for the list
**10–1F**  0

## CFNF entry for event notification

**Entry: CFNF**
**VIT option:**
    CFS
**Event:** Event Notification Facility (ENF) Exit Driven for Event X'35'
**VIT processing module:**
    ISTRACCF
**Control is returned to:**
    ISTFSXEF

This trace record is written when a change in coupling facility resources or in the coupling facility structure has occurred that may now allow a connection to a structure that had previously failed.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| CFNF | I D | 0 | F U N C | 0 | CFSSTR ADDRESS | STRUCTURE NAME | RPH ADDRESS |

**Byte (hex)**
  **Contents**
**00–03**  Record ID: C"CFNF"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  0
**06**  Exit function codes pertaining to ENF Events with event code X'35':
  **X'01'**  A coupling facility resource has become available that may allow a structure to successfully connect.
  **X'02'**  A specific coupling facility structure has become available.
**07**  0
**08–0B**  Structure object (CFSSTR) address
**0C–1B**  If the exit function code is X'02', this will contain the structure name. Otherwise, it will contain 0.
**1C–1F**  Request parameter header (RPH) address

# CFPG entry for IXLPURGE service

**Entry:  CFPG**
**VIT option:**
  CFS
**Event:**  Invocation of the MVS macro IXLPURGE
**VIT processing module:**
  ISTRACCF
**Control is returned to:**
  ISTFSNPG

This trace record is written when VTAM issues an IXLPURGE macro to purge outstanding accesses to the coupling facility structure.

| 0 0 0 0 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CFPG | ID | 0 | | FLAGS | CFSSTR ADDRESS | | | | 0 | | | | RETURN ADDRESS | | | | MVS RETURN CODE | | | | MVS REASON CODE | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"CFPG"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–06**  0

**07**  Event flags:

**Bit**  **Meaning**

**1... ....**
Purged by request ID (REQID)

**.1.. ....**
Purged by address space token (STOKEN)

**..1. ....**
Purged by task token (TTOKEN)

**...x xxxx**
0

**08–0B**  Structure object (CFSSTR) address.

**0C–0F**  0

**10–13**  Address of the invoker of IXLPURGE.

**14–17**  Return code from the MVS macro IXLPURGE.

**18–1B**  Reason code from the MVS macro IXLPURGE.

**1C–1F**  Request parameter header (RPH) address.

# CFP2 entry for IXLPURGE service (Part 2)

**Entry:  CFP2**

**VIT option:**
CFS

**Event:**  Invocation of the MVS macro IXLPURGE

**VIT processing module:**
ISTRACCF

This trace record is a continuation of the CFPG entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CFP2 | 0 | REQUEST IDENTIFIER, ADDRESS SPACE TOKEN OR TASK TOKEN |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFP2"

**04–0F** 0

**10–1F** If purging by request identifier, the 8-byte request identifier, padded to the right with zeros.

If purging by address space, the 8-byte address space token, padded to the right with zeros.

If purging by task, the 16-byte address space token.

# CFRB entry for structure rebuild service

**Entry: CFRB**

**VIT option:**

CFS

**Event:** Invocation of the MVS macro IXLREBLD

**VIT processing module:**

ISTRACCF

**Control is returned to:**

ISTFSURB

This trace record is written when a rebuild of the coupling facility structure is started, stopped, or completed, using the IXLREBLD macro.

| 0 0 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CFRB | ID | FLAGS | FUNCTION | CONNECTION ID | CFSSTR ADDRESS | CONNECT REASON CODE | INVOKER ADDRESS | MVS RETURN CODE | MVS REASON CODE | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"CFRB"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Macro Parameter Flags

| | Bit | Meaning |
|---|---|---|
| | **1... ....** | |
| | | STARTREASON(LOSSCONN) was specified for the request. |
| | **.1.. ....** | |
| | | STARTREASON(STRFAILURE) was specified for the request. |
| | **..1. ....** | |
| | | STOPREASON(LOSSCONNOLD) was specified for the request. |
| | **...1 ....** | |
| | | STOPREASON(LOSSCONNEW) was specified for the request. |
| | **.... 1...** | |
| | | STOPREASON(STRFAILUREOLD) was specified for the request. |
| | **.... .1..** | |
| | | The rebuild was started or stopped for a connection-specific reason. Bytes X'0C'–X'0F' contain the connection-specific reason. |
| | **.... ..xx** | |
| | | 0 |
| 06 | | The function requested by this invocation of IXLREBLD: |
| | **X'01'** | REQUEST(START) was specified. |
| | **X'02'** | REQUEST(STOP) was specified. |
| | **X'03'** | REQUEST(COMPLETE) was specified. |
| 07 | | Connection identifier of the invoker of IXLREBLD |
| 08–0B | | Structure object (CFSSTR) address |
| 0C–0F | | Connection-specific reason for starting or stopping the rebuild. |
| 10–13 | | The address of the invoker of IXREBLD |
| 14–17 | | Return code for the MVS macro IXLREBLD |
| 18–1B | | Reason code for the MVS macro IXLREBLD |
| 1C–1F | | Request parameter header (RPH) address |

# CFTP entry for coupling facility TCP/IP interface requests

**Entry: CFTP**
**VIT option:**
CFS
**Event:** Completion of a request from TCP/IP to CFS
**VIT processing module:**
ISTRACCF
**Control is returned to:**
ISTFSDPC or ISTFSVPC

This trace record is written when a request from TCP/IP to the coupling facility completes.

| 0 0 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| CFTP | ID | FUNCTION | OPERATION | FLAGS OBJECT ADDRESS | CFUSR ADDRESS | TCP NAME | RETURN CODE | RPH ADDRESS |

**Byte (hex)**
**Contents**

| | |
|---|---|
| **00–03** | Record ID: C"CFTP" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | The TCP/IP function identifier: |

    **X'01'**    Sysplex Wide Security Associations
    **X'02'**    Sysplexports

| | |
|---|---|
| **06** | The operation requested by TCP/IP for the following function: |

    Sysplex Wide Security Associations
    **X'01'**    RegisterUser
    **X'02'**    DeregisterUser
    **X'03'**    ClaimList
    **X'04'**    UpdateEntry
    **X'05'**    DeleteEntry
    **X'06'**    DeleteMult
    **X'07'**    FreeList
    **X'08'**    ReceiveData
    **X'09'**    RepopulateComplete
    **X'0A'**    QuiesceComplete
    **X'0B'**    InitSeq#
    **X'0C'**    GetSeq#
    **X'0D'**    QuerySeq#
    **X'0E'**    ValidateList
    **X'0F'**    FreeAll
    Sysplexports
    **X'01'**    RegisterUser
    **X'02'**    DeregisterUser
    **X'03'**    AssociateStack
    **X'04'**    GetEphemeralPort
    **X'05'**    MarkEphemeralPort
    **X'06'**    UnassociateStack
    **X'07'**    FreeList
    **X'08'**    QueryList
    **X'09'**    RepopulateComplete
    **X'0A'**    QuiesceComplete

| | |
|---|---|
| **07** | TCP/IP request flags |

**Bit**    **Meaning**
**1... ....**
    Request issued as part of repopulation processing
**.xxx xxxx**
    0

| | |
|---|---|
| **08–0B** | Structure object address |
| **0C–0F** | CFS TCP/IP user control block (CFUSR) address |
| **10–17** | The TCP/IP stack name |
| **18–1B** | Return code for the request |
| **1C–1F** | Request parameter header (RPH) address |

## CFT2 entry for coupling facility TCP/IP interface requests (Part 2)

**Entry:**  **CFT2**
**VIT option:**
    CFS
**Event:**  Completion of a request from TCP/IP to CFS
**VIT processing module:**
    ISTRACCF

This trace record is a continuation of the CFTP entry. As many of these as necessary are generated to display the entire parameter list.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CFT2 | PARAMETER LIST DATA (UP TO 28 BYTES) |

**Byte (hex)**

**Contents**

00–03   Record ID: C"CFT2"

04–1F   28 bytes of the parameter list data for this request

# CFTX entry for coupling facility TCP/IP event exit invocation

**Entry:  CFTX**

**VIT option:**

CFS

**Event:**  Invocation of the CFS TCP/IP event exit

**VIT processing module:**

ISTRACCF

**Control is returned to:**

ISTFSNTX

This trace record is written when an asynchronous event causes CFS to drive the TCP/IP event exit routine.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| CFTX | I D | F U N C T Y P E | E V E N T C O D E | R T N C O D E | STRUC. OBJECT ADDRESS | REQUEST ID | TCP NAME | DATA BUFFER ADDRESS | RPH ADDRESS |

**Byte (hex)**

**Contents**

00–03   Record ID: C"CFTX"

04       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05       The TCP/IP function identifier:

**X'01'**   Sysplex Wide Security Associations

**X'02'**   Sysplexports

06       The event code describing the asynchronous event:

**X'01'**   Quiesce

|  |  |
|---|---|
| X'02' | Repopulate/Rebuild |
| X'03' | Repopulate/Reconnect |
| X'04' | Deregistered |
| X'05' | Connection Attempt Failed |
| X'06' | Received Data |
| X'07' | DeleteMult Complete |
| X'08' | FreeList Complete |
| X'09' | FreeAll Complete |

**07**    Return code:

|  |  |
|---|---|
| X'00' | Success |
| X'08' | Failure |

**08–0B**  Structure object address

**0C–0F**  Request Correlation ID - Correlates this event with the original request

**10–17**  The TCP/IP stack name

**18–1B**  Data buffer address, if event is "Received Data"; otherwise 0

**1C–1F**  Request parameter header (RPH) address

# CFUS entry for user sync point service

**Entry: CFUS**

**VIT option:**
    CFS

**Event:** Invocation of the MVS macro IXLUSYNC

**VIT processing module:**
    ISTRACCF

**Control is returned to:**
    ISTFSUUS

This trace record is written when a connector sets or confirms a user sync point for a coupling facility structure with the IXLUSYNC macro.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| CFUS | ID | 0 | FUNC | CONNECT ID | CFSSTR ADDRESS | USER EVENT | RETURN ADDRESS | RETURN CODE | REASON CODE | RPH ADDRESS |

**Byte (hex)**

    **Contents**

**00–03**  Record ID: C"CFUS"

**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**    0

**06**    Function requested by this invocation of IXLUSYNC:

|  |  |
|---|---|
| X'01' | SET function requested |
| X'02' | CONFIRM function requested |
| X'03' | CONFIRMSET function requested |

**07**    Connection identifier of the invoker of IXLUSYNC

08–0B   Structure object (CFSSTR) address
0C–0F   The value of USEREVENT parameter for this invocation of IXLUSYNC
10–13   The address of the invoker of IXLUSYNC
14–17   Return code for the MVS macro IXLUSYNC
18–1B   Reason code for the MVS macro IXLUSYNC
1C–1F   Request parameter header (RPH) address

# CFVC entry for IXLVECTR service

**Entry:  CFVC**
**VIT option:**
    CFS
**Event:**  Invocation of the MVS macro IXLVECTR
**VIT processing module:**
    ISTRACCF
**Control is returned to:**
    ISTFSLVC

This trace record is written when VTAM issues an IXLVECTR macro to modify or test the list notification vector.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| CFVC | ID | 0 | FUNCTION | 0 | CFSSTR ADDRESS | VECTOR INDEX | RETURN ADDRESS | RETURN CODE | BIT STRING OR ACTUAL LENGTH | RPH ADDRESS |

**Byte (hex)**
        **Contents**
00–03   Record ID: C"CFVC"
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05      0
06      Function codes
        **X'01'**   Modify vector size
        **X'02'**   Load and test a range of vector entries
        **X'03'**   Test a single list state.
07      0
08–0B   Structure object (CFSSTR) address.
0C–0F   If byte 06 indicates a modify vector size, then this field contains the new length of the vector.

        If byte 06 indicates a load and test function, then this field contains the starting vector index to begin testing. Thirty-two consecutive bits will be tested.

        If byte 06 indicates a test function, then this field contains the vector index being tested.
10–13   Address of the invoker of IXLVECTR.

**14–17** Return code from the MVS macro IXLVECTR. Refer to the *z/OS MVS Programming: Sysplex Services Reference* for an explanation of this code.

**18–1B** If byte 06 indicates a modify vector size function, then this field contains the actual length of the new vector.

If byte 06 indicates a load and test function, then this field contains a 32-bit string where each bit indicates the following:
**0** List is not empty
**1** List is empty

**1C–1F** Request parameter header (RPH) address.

---

# CHGO entry for CHANGE_OWNER requests

**Entry: CHGO**
**VIT option:**
    CSM
**Event: IVTCSM REQUEST=CHANGE_OWNER**
**VIT processing module:**
    ISTITCCS
**Control is returned to:**
    IVTSMCCO

This trace record provides the status of an IVTCSM REQUEST=CHANGE_OWNER macroinstruction.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 | 0A 0B | 0C 0D | 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CHGO | ID | 0 | FLAG | RECNUM | 0 | OWNER | RTN COD | RSN COD | UTILRTN CALLER OR RETURN ADDRESS | LAST BUFFER LIST ENTRY | NUMBER OF BUFFERS | THREAD VALUE OR 0 |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"CHGO"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Trace record flag:

    **B'0... ....'**
        Indicates that this is the last record for this event.

    **B'1... ....'**
        Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.
**07** Trace record number to correlate all the entries for this particular event.
**08–09** 0
**0A–0B** Owner ID
**0C–0D**
        Return code
**0E–0F** Reason code

**10–13** Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

**14–17** Last buffer list entry

**18–1B** Number of buffers

**1C–1F** THREAD value if specified or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

## CHG2 entry for CHANGE_OWNER requests

**Entry: CHG2**

**VIT option:**
   CSM

**Event:** IVTCSM REQUEST=CHANGE_OWNER

**VIT processing module:**
   ISTITCCS

This trace record is a continuation of the CHGO trace record. Each CHG2 record traces, at most, two buffers that were requested for ownership change.

| 0 0 0 0 | 0 0 0 | 0 | 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 | 7 | 8 9 A B C D E F 0 1 2 3 | 4 5 6 7 8 9 A B C D E F |
| CHG2 | 0 | F L A G / R E C N U M | BUFFER TOKEN FOR INPUT BUFFER LIST ENTRY | BUFFER TOKEN FOR INPUT BUFFER LIST ENTRY OR 0 |

**Byte (hex)**

   **Contents**

**00–03** Record ID: C"CHG2"

**04–05** 0

**06** Trace record flag:

   **B'0... ....'**
      Indicates that this is the last record for this event.

   **B'1... ....'**
      Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07** Trace record number to correlate all the entries for this particular event

**08–13** Buffer token contained in input buffer list entry

**14–1F** Buffer token contained in input buffer list entry or 0

## CI1 or CO1 trace entries

This trace record provides information about the inbound and outbound requests sent to session services to begin or end a same-network or cross-network LU-LU session.

The CI1 and CO1 traces have two formats:

- **Format 0** is used for same-network sessions or when VTAM does not know whether the request is for a cross-network session. This format contains parts 1, 2, and 3 of the CIn or COn record.
- **Format 1** is used for cross-network sessions. This format contains parts 1, 2, 3, and 4 of the CIn or COn record.

# CI1 or CO1 entry for SSCP (RUPE — Part 1)

**Entry:  CI1 or CO1**
**VIT option:**
     SSCP
**Event:  Requests (Part 1)**
**VIT processing module:**
     ISTRACSC
**Control is returned to:**
     Module invoking the INTRACE macro that caused the record to be produced

Part 1 of the CIn or COn record is for both format 0 and format 1 and contains essentially the same information as the CCI or CCO entry.

If this entry is associated with an event failure (that is, the sense data is nonzero), this entry is generated whether the SSCP option is in effect or not. It is treated as an exception condition and, therefore, is traced whenever the VIT is active.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CI1 OR CO1 | | | FLAGS | IDD | CBID | | ITFSM | | SAVE AREA ADDRESS | | | SAVE AREA ID | | | | RETURN ADDRESS | | | | ORIGIN NETWORK ADDRESS | | | | | | DESTINATION NETWORK ADDRESS | | | | | |

**Byte (hex)**
     **Contents**
**00–02**  Record ID:
      - C"CI1" for inbound processing
      - C"CO1" for outbound processing
**03**   Flags

      **Bit       Meaning**
      **0... ....**
              Requests
      **1... ....**
              Response
      **.0.. ....**
              No sense
      **.1.. ....**
              Sense
      **.... ..00**
              Format 0
      **.... ..01**
              Format 1

| | |
|---|---|
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | Control block ID=X'54' |
| 06–07 | Initiation and termination finite state machines (taken from the session information block, SIB) |
| 08–0B | Save area address |
| 0C–0F | Save area ID (bytes 4, 5, 7, and 8 of the module name if available) |
| 10–13 | Address of the issuer of the macro (CPCALL, CPEXIT, or CPWAIT) |
| 14–19 | Network address at origin (RUPEOAF) |
| 1A–1F | Network address at destination (RUPEDAF) |

# CI1 or CO1 entry for SSCP (NCSPL — Part 1)

**Entry: CI1 or CO1**
**VIT option:**
   SSCP
**Event:** Requests (Part 1)
**VIT processing module:**
   ISTRACSC
**Control is returned to:**
   Module invoking the INTRACE macro that caused the record to be produced

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CI1 OR CO1 | | | FLAGS | ID | CBID | FLAGS | RTNCD | SAVE AREA ADDRESS | | | | SAVE AREA ID | | | | RETURN ADDRESS | | | | CPCB OPCODE | | | | TYPE | 0 | DESTINATION NETWORK ADDRESS | | | | | |

**Byte (hex)**
   **Contents**
**00–02** Record ID:
   - C"CI1" for inbound processing
   - C"CO1" for outbound processing
**03** Flags

   | Bit | Meaning |
   |---|---|
   | `0... ....` | Requests |
   | `1... ....` | Response |
   | `.0.. ....` | No sense |
   | `.1.. ....` | Sense |
   | `.... ..00` | Format 0 |
   | `.... ..01` | Format 1 |

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Control block ID=X'60'

**06** Flags (CPCBFL)
**07** CPCB return code (CPCBRC)
**08–0B** Save area address
**0C–0F** Save area ID (bytes 4, 5, 7, and 8 of the module name if available)
**10–13** Address of the issuer of the macro (CPCALL, CPEXIT, or CPWAIT)
**14–17** CPCB operation code (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)
**18** WTD type
**19** 0
**1A–1F** Network address at destination (NCSPLDAF)

# CI1 or CO1 entry for SSCP (not RUPE or NCSPL — Part 1)

**Entry:** CI1 or CO1
**VIT option:**
     SSCP
**Event:** Requests (Part 1)
**VIT processing module:**
     ISTRACSC
**Control is returned to:**
     Module invoking the INTRACE macro that caused the record to be produced

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CI1 OR CO1 | | | FLAGS | IDB ID | CBID | CPCBF | RTNCD | SAVE AREA ADDRESS | SAVE AREA ID | RETURN ADDRESS | CPCB OPCODE | CPCB WORD 3 | CPCB WORD 4 |

**Byte (hex)**
     **Contents**
**00–02** Record ID:
     • C"CI1" for inbound processing
     • C"CO1" for outbound processing
**03** Flags

     | Bit | Meaning |
     |---|---|
     | **0... ....** | Requests |
     | **1... ....** | Response |
     | **.0.. ....** | No sense |
     | **.1.. ....** | Sense |
     | **.... ..00** | Format 0 |
     | **.... ..01** | Format 1 |

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)
**06** CPCB flag (CPCBFL)
**07** Return code (CPCBRC)
**08–0B** Save area address
**0C–0F** Save area ID (bytes 4, 5, 7, and 8 of the module name if available)
**10–13** Address of the issuer of the macro (CPCALL, CPEXIT, or CPWAIT)
**14–17** CPCB operation code (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)
**18–1B** CPCB word 3 (field WTDPTR)
**1C–1F** CPCB word 4 (field CPCBPH)

# CI2 or CO2 entry for SSCP (RUPE — Part 2)

**Entry:** CI2 or CO2
**VIT option:**
    SSCP
**Event:** Requests (Part 2)
**VIT processing module:**
    ISTRACSC

The CI2 and CO2 trace records are continuations of the CI1 and CO1 entries.

| 0 0 0 | 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| CI2 OR CO2 | C B I D | FIRST 24 BYTES OF RU | SENSE DATA OR 0 |

**Byte (hex)**

    **Contents**
**00–02** Record ID:
- C"CI2" for inbound processing
- C"CO2" for outbound processing

**03** Control block ID=X'54'
**04–1B** First 24 bytes of RU
**1C–1F** Sense data or 0

# CI2 or CO2 entry for SSCP (NCSPL — Part 2)

**Entry:** CI2 or CO2
**VIT option:**
    SSCP
**Event:** Requests (Part 2)
**VIT processing module:**
    ISTRACSC

This trace record is a continuation of the CI1 or CO1 trace record.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| CI2 OR CO2 | | | C B I D | RDTE STATE | | | | CPCB WTD | | | | 0 | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
> **Contents**

**00–02** Record ID:
- C"CI2" for inbound processing
- C"CO2" for outbound processing

**03** Control block ID=X'60'

**04–07** RDTE state (in control block RPRE). This value, a resource status code, is explained in *z/OS Communications Server: IP and SNA Codes* and in control block FSM. Refer to *z/OS Communications Server: SNA Data Areas Volume 1*.

**08–0B** CPCB WTD (first four bytes)

**0C–1F** 0

## CI2 or CO2 entry for SSCP (not RUPE or NCSPL — Part 2)

**Entry: CI2 or CO2**
**VIT option:**
> SSCP

**Event:** Requests (Part 2)
**VIT processing module:**
> ISTRACSC

This trace record is a continuation of the CI1 or CO1 trace record.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| CI2 OR CO2 | | | C B I D | 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
> **Contents**

**00–02** Record ID:
- C"CI2" for inbound processing
- C"CO2" for outbound processing

**03** Control block ID (Refer to*z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)

**04–1F** 0

## CI3 or CO3 entry for SSCP (RUPE, NCSPL, or not RUPE or NCSPL — Part 3)

**Entry: CI3 or CO3**

**VIT option:**
SSCP
**Event:** Requests (Part 3)
**VIT processing module:**
ISTRACSC

Part 3 of the CIn or COn record is for both format 0 and format 1 and contains the address of the ISTSIB for the session, the ISTPCID, and the PLU and SLU names.

**Note:** If this is the last CIn or COn entry, these are the PLU and SLU names as known in this network. If this entry is followed by CI4 or CO4, these are just the PLU and SLU names. The CI4 or CO4 entry identifies the network.

| 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 | 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|---|
| CI3 OR CO3 | 0 | SIB ADDRESS | PCID FOR THIS REQUEST | PLU NAME | SLU NAME |

**Byte (hex)**
**Contents**
**00–02** Record ID:
- C"CI3" for inbound processing
- C"CO3" for outbound processing

**03** 0
**04–07** SIB address
**08–0F** PCID for this request
**10–17** PLU name
**18–1F** SLU name

# CI4 or CO4 entry for SSCP (RUPE, NCSPL, or not RUPE or NCSPL — Part 4)

**Entry:** CI4 or CO4
**VIT option:**
SSCP
**Event:** Requests (Part 4)
**VIT processing module:**
ISTRACSC

Part 4 of the CIn or COn entry is for format 1 only.

| 0 0 0 | 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 | 3 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|
| CI4 OR CO4 | 0 | PCID IN PREVIOUS REQUEST | PLU NETID | SLU NETID |

**Byte (hex)**

**Contents**

00–02 Record ID:
- C"CI4" for inbound processing
- C"CO4" for outbound processing

03–07 0

08–0F PCID in previous request

10–17 PLU network ID

18–1F SLU network ID

# CMER entry for error in CMIP services (Part 1)

**Entry: CMER**

**VIT option:**

None (Generated by CMIP services)

**Event:** Error discovered in CMIP services (Part 1)

**VIT processing module:**

ISTITCCM

**Control is returned to:**

Module invoking the INTRACE macro that caused the record to be produced

This trace record is generated when an error is detected in CMIP services. For long CMIP strings, a maximum of seven CME2 trace records are generated. For longer CMIP strings, an additional CMER trace record with a maximum of seven CME2 trace records are generated.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

This entry is always traced, regardless of the VIT options specified.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| CMER | | | | ID | 0 | ERROR | CODE | PLACE CODE | | | | DATA OR 0 | | | | | | | | | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

00–03 Record ID: C"CMER"

04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05 0

06–07 Error code indicating which error is detected

08–0B Place code indicating where the error is detected

0C–1B The first 16 bytes of data (padded on the right with zeros). 0 if there is no data. The data consists of one or more variables. Each variable is preceded by a 2-byte field indicating its length in bytes. The length of the data itself, not the number of bytes being traced, is used.

If the data is too long to fit in the VIT record, the length field does not match the number of bytes traced.

**1C–1F**  Request parameter header (RPH) address

## CME2 entry for error in CMIP services (Part 2)

**Entry:  CME2**
**VIT option:**
> None (Generated by CMIP services)

**Event:**  Error discovered in CMIP services (Part 2)
**VIT processing module:**
> ISTITCCM

This trace record is a continuation of the CMER entry. It contains 28 more bytes of the CMIP error data. A maximum of seven CME2 trace records are generated. The number of records depends on the length of the data.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CME2 | 28 BYTES OF DATA |

**Byte (hex)**
> **Contents**

**00–03**  Record ID: C"CME2"
**04–1F**  Up to 28 bytes of data (padded on the right with zeros)

## CNA entry for VCNSCMD issued by application

**Entry:  CNA**
**VIT option:**
> VCNS

**Event:**  VCNSCMD issued by application
**VIT processing module:**
> ISTRACNS

**Control is returned to:**
> ISTAICVC

This CNA trace record shows VCNS command requests. The API routine ISTAICVC writes the entry and then queues the RPL to the correct PAB.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CNA | | | | ID | EXTDS | OPT1 | CNTRL | RPL ADDRESS | | | | RPL3 ADDRESS | | | | RPL AREA | | | | VCNS FLAGS | | | 0 | OPERAND-DEPENDENT DATA | | | | | | | |

**Byte (hex)**

**Contents**

**00–03**    Record ID: C"CNA"

**04**       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**       Exit definition (RPLEXTDS)

| Bit | Meaning |
|---|---|
| **1... ....** | Indicates RPL exit was scheduled |
| **.1.. ....** | Indicates RPL exit not specified |
| **..1. ....** | Indicates RPL exit was specified |
| **...x xx..** | 0 |
| **.... ..1.** | Indicates BRANCH=YES specified |
| **.... ...x** | 0 |

**06**       Option code byte 1 (RPLOPT1)

| Bit | Meaning |
|---|---|
| **xxxx ....** | 0 |
| **.... 1...** | Asynchronous request indicator |
| **.... .xx.** | 0 |
| **.... ...1** | External ECB indicator |

**07**       VCNSCMD CONTROL= operand value

| | |
|---|---|
| **08** | INQUIRE |
| **10** | LOGON |
| **14** | TEST |
| **18** | XID |
| **20** | LOGOFF |
| **40** | STATUS |
| **50** | SET |
| **80** | SEND |
| **C0** | CHECK |
| **E0** | SETCPARM |
| **F0** | RECEIVE |
| **FC** | REPLY |

**08–0B**    RPL address

**0C–0F**    RPL3 address (RPLAAREA)

**10–13** Address of area into which data is to be read or from which data is to be written (RPLAREA), or 0 if no area is supplied

**14** VCNS flag byte

| Bit | Meaning |
|---|---|
| **1... ....** | Q-bit indicator |
| **.1.. ....** | M-bit indicator |
| **..1. ....** | D-bit indicator |
| **...1 ....** | Buffer list indicator |
| **.... xxxx** | 0 |

**15** VCNS flag byte

| Bit | Meaning |
|---|---|
| **00.. ....** | Continue specific |
| **01.. ....** | Continue any |
| **11.. ....** | Continue same |
| **..00 ....** | Data flow=on |
| **..01 ....** | Data flow=off |
| **..11 ....** | Data flow=same |
| **.... 1...** | Receive any indicator |
| **.... .xxx** | 0 |

**16** VCNS flag byte

| Bit | Meaning |
|---|---|
| **1... ....** | STYPE=CONFIRM indicator |
| **.xxx xxxx** | 0 |

**17** 0

**18–1F** When CONTROL equals LOGON or CONTROL equals INQUIRE: symbolic name of the network access point

**18–1B** When CONTROL does not equal LOGON and CONTROL does not equal INQUIRE and the request is an X.25 request: address into which expedited data is to be read or from which expedited data is to be written, or 0 if no area is supplied

**1C–1F** When CONTROL does not equal LOGON and CONTROL does not equal INQUIRE: connection or resource identifiers (RID)
  - RID is supplied when the VCNSCMD CONTROL value is:
      LOGOFF
      OPEN (STYPE=REQUEST)
      READ
      RECEIVE (when SMODE is CONNECTIONLESS)
      REPLY
      SEND (when the connection ID value is 0)

SET
SETCPARM (when the connection ID value is 0)
STATUS (when the RID value is not 0)
TEST
XID

- Connection identifier is supplied when the VCNSCMD CONTROL value is:

  CLOSE
  EXPEDITE
  OPEN (STYPE=CONFIRM)
  RECEIVE (Specific)
  RESET
  RESUME
  SEND (when the connection ID value is not 0)
  SETCPARM (when the connection ID value is not 0)
  SUSPEND
  STATUS (when the RID value is 0)

# CNP1 or CNR1 entry for ECB posted or RPL exit dispatched (Part 1)

**Entry:  CNP1 or CNR1**
**VIT option:**
  VCNS
**Event:  ECB posted or RPL exit dispatched (Part 1)**
**VIT processing module:**
  ISTRACNS
**Control is returned to:**
  ISTAICPT for CNP1. ISTAPCUE for CNR1

The CNP1 trace record is written when an ECB is posted. The CNR1 trace record is written when an RPL exit is dispatched. These trace records signal that execution of the VCNSCMD macroinstruction is complete and show the data returned to the user application program.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 | 17 | 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| CNP1 OR CNR1 | ID | RTNCD | FDB2 | CNTRL | RPL ADDRESS | RPL3 ADDRESS | RPL AREA | VCNS FLAGS | 0 | PARAMETER-DEPENDENT DATA |

**Byte (hex)**
  **Contents**
**00–03**  Record ID:
  - C"CNP1" for ECB posting
  - C"CNR1" for RPL exit dispatching
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  Global VTAM return code (RPLRTNCD)
**06**  Global VTAM feedback code (RPLFDB2)
**07**  VCNSCMD CONTROL= operand value
  **08**  INQUIRE

|      |          |
|------|----------|
| **10** | LOGON |
| **14** | TEST |
| **18** | XID |
| **20** | LOGOFF |
| **40** | STATUS |
| **50** | SET |
| **80** | SEND |
| **C0** | CHECK |
| **E0** | SETCPARM |
| **F0** | RECEIVE |

**08–0B**  RPL address

**0C–0F**  RPL3 address (RPLAAREA)

**10–13**  Address of area containing user data (RPLAREA), or 0 if no area is supplied

**14**  VCNS flag byte

| **Bit** | **Meaning** |
|---------|-------------|
| **1... ....** | Q-bit indicator |
| **.1.. ....** | M-bit indicator |
| **..1. ....** | D-bit indicator |
| **...1 ....** | Buffer list indicator |
| **.... xxxx** | 0 |

**15**  VCNS flag byte

| **Bit** | **Meaning** |
|---------|-------------|
| **00.. ....** | Continue specific |
| **01.. ....** | Continue any |
| **11.. ....** | Continue same |
| **..00 ....** | Data flow=on |
| **..01 ....** | Data flow=off |
| **..11 ....** | Data flow=same |
| **.... 1...** | Receive any indicator |
| **.... .xxx** | 0 |

**16**  VCNS flag byte

| **Bit** | **Meaning** |
|---------|-------------|
| **1... ....** | STYPE=CONFIRM indicator |
| **.xxx xxxx** | 0 |

**17**  0

**18–1B**  Address into which expedited data is to be read or from which expedited data is to be written, or 0 if no area is supplied

**1C–1F**  Connection or resource identifier (RID)

- RID is returned when the VCNSCMD CONTROL value is:
  LOGON
  LOGOFF
  READ
  SET
  STATUS (for network access point)
  TEST
  XID
- Connection ID is returned when the VCNSCMD CONTROL value is:
  CLOSE
  EXPEDITE
  OPEN
  RECEIVE
  RESET
  RESUME
  SEND
  SETCPARM
  SUSPEND
  STATUS (for connection)
- 0 is returned when the VCNSCMD CONTROL value is INQUIRE

## CNP2 or CNR2 entry for ECB posted or RPL exit dispatched (Part 2)

**Entry: CNP2 or CNR2**
**VIT option:**
    VCNS
**Event:** ECB posted or RPL exit dispatched (Part 2)
**VIT processing module:**
    ISTRACNS

The CNP2 trace record is a continuation of the CNP1 trace record. The CNR2 trace record is a continuation of the CNR1 trace record.

| 0 0 0 0 | 0 0 0 0 | 0 0 | 0 0 | 0 0 0 0 | 1 1 | 1 1 | 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 | A B | C D E F | 0 1 | 2 3 | 4 5 6 7 | 8 9 A B C D E F |
| CNP2 OR CNR2 | ECB ADDRESS OR EXIT ADDRESS | R C P R I | R C S E C | LENGTH OF USER DATA AREA | M A X L N | D A T L N | SENSE DATA OR 0 | EXPEDITED DATA |

**Byte (hex)**
**Contents**
**00–03** Record ID:
- C"CNP2" for continuation of CNP1
- C"CNR2" for continuation of CNR1

**04–07** ECB or RPL exit address (RPLECB)
**08–09** VCNSCMD primary return code
**0A–0B** VCNSCMD secondary return code
**0C–0F** Length of area containing user data (RPLRLEN)
**10–11** Length of expedited data area
**12–13** Length of expedited data received
**14–17** Four bytes of sense data, or 0 when no sense data is returned

18–1F    Eight bytes of expedited data

## CNP3 or CNR3 entry for ECB posted or RPL exit dispatched (Part 3)

**Entry:   CNP3 or CNR3**
**VIT option:**
       VCNS
**Event:**  ECB posted or RPL exit dispatched (Part 3)
**VIT processing module:**
       ISTRACNS

The CNP3 trace record is a continuation of the CNP2 trace record. The CNR3 trace record is a continuation of the CNR2 trace record. These records are written only when the VCNSCMD CONTROL operand value is RECEIVE or CLOSE, and RPLRLEN is greater than 0.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---------|---------|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CNP3 OR CNR3 | DATA FIELD |

**Byte (hex)**
       **Contents**
**00–03**    Record ID:
          • C"CNP3" for continuation of CNP2
          • C"CNR3" for continuation of CNR2
**04–1F**    Twenty-eight bytes of user data (RPLAREA)

## CNTP entry for buffer pool contraction

**Entry:   CNTP**
**VIT option:**
       CSM
**Event:**  Pool contraction
**VIT processing module:**
       ISTITCCS
**Control is returned to:**
       IVTSMCEX

This trace record is written when CSM pool contraction occurs.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CNTP | I D | 0 | POOL ADDRESS | EXTENT ADDRESS | NUMBER OF BUFFERS IN EXTENT | TOTAL BUFFERS AFTER CNTRACT | FREE BUFFERS AFTER CNTRACT | 0 |

**Byte (hex)**

   **Contents**
00–03   Record ID: C"CNTP"
04   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05–07   0
08–0B   Pool address
0C–0F   Extent address
10–13   Number of buffers in extent
14–17   Total number of buffers in the pool after contraction
18–1B   Number of free buffers in the pool after contraction
1C–1F   0

# CONT entry for buffer pool contraction

**Entry:   CONT**
**VIT option:**
   SMS
**Event:**   Buffer pool contraction
**VIT processing module:**
   ISTRACSM
**Control is returned to:**
   ISTORAPX

This trace record is generated whenever VTAM needs to contract a buffer pool. Buffer pool contraction will occur when VTAM determines that it has excess buffers that can be returned to the operating system.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CONT | I D | 0 | BPCB ADDRESS | PXB ADDRESS | EXTENT ADDRESS | Q U E U E | 0 | TOTAL NUMBER OF BUFFERS | AVAIL NUMBER OF BUFFERS |

**Byte (hex)**

   **Contents**
00–03   Record ID: C"CONT"
04   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05–07   0
08–0B   Address of buffer pool control block (BPCB) for which contraction occurred

0C–0F  Address of pool expansion block (PXB) that is contracted
10–13  Extent address (PXBSTADR)
14–15  Number of queued RPHs waiting for expansion
16–17  0
18–1B  Total number of buffers in pool after this contraction (BPCBTOTL)
1C–1F  Total number of available buffers in pool after this contraction

## COPY entry for internal trace statistics (Part 1)

**Entry: COPY**
**VIT option:**
> None (Created only when the trace data space is used.)

**Event:** Trace entries are copied from the fixed ECSA table to the trace data space
> ISTITDS1.

**VIT processing module:**
> ISTITCCD writes this entry directly to the data space table and only to the data space table. It is not written externally.

**Control is returned to:**
> ISTITCCD

This entry is not associated with any VIT options, but is recorded whenever the fixed ECSA table is copied to the pageable data space table.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COPY | | | | ID | | 0 | | SIZE | | | | TIME OF DAY | | | | | | | | BEGIN | | | | ENV | AMOUNT | | | EYE CATCHER | | | | NEXT COPY RECORD | | | |

**Byte (hex)**
> **Contents**
00–03  Record ID: C"COPY"
04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05  0
06-07  Size of the ECSA VIT table, in pages
08–0F  TOD, the time of day clock value when the trace entry was written
10–13  Beginning address from where the trace was copied
14  Env is an indicator of whether the copy occurred as an in-line call or under a separate SRB.

- C'I' for an in-line call

- C'S' for SRB

15–17  Amount in bytes of trace copied to the data space
18–1B  The eyecatcher GAP when a modify command has changed the SIZE of the ECSA trace table and trace records were lost during the change.

> The eyecatcher LOST when the ECSA table has wrapped before all entries could be copied to the data space table.

> If neither of these conditions apply, the field is set to blanks (X'40404040').
1C–1F  Address of the next COPY record in the data space

## COP2 entry for internal trace statistics (Part 2)

**Entry: COP2**

**VIT option:**
None (Created only when the trace data space is used.)

**Event:** Trace entries are copied from the fixed ECSA table to the trace data space ISTITDS1.

**VIT processing module:**
ISTITCCD writes this entry directly to the data space table and only to the data space table. It is not written externally.

This trace record is a continuation of the COPY entry. It is used to present time-of-day clock values that were recorded when the first trace entry on each 16-page segment in the ECSA trace table was written to that table. There will be enough COP2 records to contain one time-of-day value for each 16-page segment of the ECSA trace table from which records are being copied to the trace table in data space, as expressed in the COPY record.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|
| COP2 | ECSA VIT ADDRESS | TIME OF DAY | TIME OF DAY OR ZERO | TIME OF DAY OR ZERO |

**Byte (hex)**
**Contents**
**00–02** Record ID: C"COP2"
**04-07** Address of the beginning of the ECSA VIT table from which the copy was done.
**08-0F** If this is the first COP2 continuation record, the time-of-day clock value saved when the first trace entry was written to the 16-page segment of the ECSA trace table that contains the first entry copied to the pageable data space table. Otherwise, the time-of-day clock value saved when the first trace entry was written to the next 16-page segment of the ECSA trace table.
**10-17** If at least one trace entry was copied from the next 16-page segment of the ECSA trace table, the time-of-day clock value saved when the first trace entry was written to that 16-page segment of the ECSA trace table. Otherwise, 0.
**18-1F** If at least one trace entry was copied from the next 16-page segment of the ECSA trace table, the time-of-day clock value saved when the first trace entry was written to that 16-page segment of the ECSA trace table. Otherwise, 0.

# CPx entry for requests/responses processed by the CP (Part 1)

**Entry:** CPI or CPO
**VIT option:**
SSCP
**Event:** Requests/responses with a RUPE
**VIT processing module:**
ISTRACSC

**Control is returned to:**
    Many modules possible

This trace record provides information about inbound and outbound requests and responses that are processed by the CP.

If this entry is associated with an event failure (that is, the sense data is nonzero), this entry is generated whether the SSCP option is in effect or not. It is treated as an exception condition, and therefore, is traced whenever the VIT is active.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 | 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CPI CPO | F L A G S | I D | C B I D | 0 | | SAVE AREA ADDRESS | SAVE AREA ID | RETURN ADDRESS | RUPE ORIGIN NETWORK ADDRESS | RUPE DESTINATION NETWORK ADDRESS |

**Byte (hex)**
    **Contents**
**00–02** Record ID:

- C"CPI" for inbound processing

- C"CPO" for outbound processing

**03** Flags

- X'80' = response RU

- X'40' = sense traced

- X'00' = request RU

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control Block ID=X'54'

**06–07** 0

**08–0B** Save Area Address

**0C–0F** Save Area ID

**10–13** Address of the issuer of the APSEND macro

**14–19** For sender, network address at origin

**1A–1F** For sender, network address at destination

# CP2 entry for requests/responses processed by the CP (Part 2)

**Entry:** CP2

**VIT option:**
    SSCP

**Event:** Requests/responses with a RUPE (Part 2)

**VIT processing module:**
    ISTRACSC

This trace record is a continuation of the CPI or CPO entry.

| 00 | 01 | 02 | 03 | 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B | 1C 1D 1E 1F |
|----|----|----|----|----|----|
| CP2 | | | CBID | FIRST 24 BYES OF RU | SENSE CODE OR 0 |

**Byte (hex)**

    **Contents**

**00–02**    Record ID: C"CP2"

**03**    Control Block ID=X'54'

**04–1B**    First 24 bytes of RU

**1C–1F**    Sense code or 0

# CPPG or CPPT entry for CPPURGE or CPPOST event (Part 1)

**Entry: CPPG or CPPT**

**VIT option:**

    SSCP

**Event: CPPURGE or CPPOST Macro (Part 1)**

**VIT processing module:**

    ISTRACSC

**Control is returned to:**

    ISTPVCWP

This trace record gives information about CPPURGE or CPPOST macroinstruction processing. It may help you determine why I/O is outstanding or why an SSCP request is failing.

CPPG and CPPT are the complements of CPWT. Each CPPT is preceded by a CPWT. But a CPPG can post several events at once, as it may be concluding several CPWT entries.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|----|----|----|----|----|----|----|----|----|----|----|
| CPPG OR CPPT | ID | RTNCD | TYPE | OPT | LQAB GROUP ADDRESS | WREDATA IN WRE | WORK ELEMENT ADDRESS | OPC IN WORK ELEMENT OR 0 | SAVE AREA ID OR 0 | SENSE DATA OR 0 |

**Byte (hex)**

    **Contents**

**00–03**    Record ID:

      • C"CPPG" for CPPURGE processing

      • C"CPPT" for CPPOST processing

**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**    Return code set by CPPOST or CPPURGE

**06**      Type flags

     **Bit**      **Meaning**

     **.... ...x**
         1 = PVI event

     **.... ..x.**
         1 = search on EID

         0 = search on URC

     **.... .x..**
         For OPTIONS(POST) only, 1 indicates that return of a control block is optional. This bit corresponds to the CTLBLKOP keyword on the CPPOST macro.

     **.... x...**
         For OPTIONS(FIND) only, 1 indicates that the invoker intends to change the WRE's event ID. This bit corresponds to the CHGEID keyword on the CPPOST macro.

**07**      High-order 4 bits: Type of LQAB used for the search

     **Value**      **LQAB type**
     **0000**      Global LQAB
     **0001**      EID-related LQAB
     **0010**      DAF-related LQAB
     **0011**      URC-related LQAB

     Low-order 4 bits: Function requested through the OPTIONS keyword on the CPPOST macro, 0 for CPPG

     **Value**      **Keyword**
     **0000**      FIND
     **0001**      DEQ
     **0010**      POST

**08–0B**      LQAB group address
**0C–0F**      Data field in ISTWRE
**10–13**      Work element address
**14–17**      CPCB operation code that indicates the original request for this work element or 0 (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)
**18–1B**      For PVI events, caller's save area ID or 0. For non-PVI events, save area ID
**1C–1F**      For POST or PURGE, sense data or 0. For DEQ or FIND, 0

# CPP2 entry for CPPURGE or CPPOST (Part 2)

**Entry:**   CPP2
**VIT option:**
     SSCP
**Event:**   CPPURGE or CPPOST macro (Part 2)
**VIT processing module:**
     ISTRACSC

This trace record contains part or all of the RU for CPPURGE or CPPOST macroinstruction processing. It may help you determine why I/O is outstanding or why an SSCP request is failing. If an RU is fewer than 24 bytes long, only the bytes in the actual RU will be moved to the corresponding fields in the trace records. The leftover bytes will be set to 0.

**Note:** This trace record is produced only for PVI events where a RUPE is available.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CPP2 | RETURN ADDRESS | FIRST 24 BYTES OF RU OR 0 |

**Byte (hex)**
**Contents**
**00–03**  Record ID: C"CPP2"
**04–07**  Return Address
**08–1F**  First 24 bytes of RU or 0

# CPP3 entry for CPPURGE or CPPOST (Part 3)

**Entry:**  CPP3
**VIT option:**
SSCP
**Event:**  CPPURGE or CPPOST macro (Part 3)
**VIT processing module:**
ISTRACSC

This trace record is a continuation of the CPPG or CPPT trace record.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C | D E F |
| CPP3 | 25 BYTES OF URC | 0 |

**Byte (hex)**
**Contents**
**00–03**  Record ID: C"CPP3"
**04–1C**  25 bytes of URC
**1D–1F**  0

# CPP4 entry for CPPURGE or CPPOST (Part 4)

**Entry:**  CPP4
**VIT option:**
SSCP
**Event:**  CPPURGE or CPPOST macro (Part 3)
**VIT processing module:**
ISTRACSC

This trace record contains part or all of the event ID (EID) for CPPURGE or CPPOST macroinstruction processing. If an EID is fewer than 24 bytes long, only the bytes in the actual EID will be moved to the corresponding fields in the trace records. The leftover bytes will be set to 0.

**Note:** This trace record is produced only when the WRE points to an EID.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CPP4 | 0 | E I D L N | FIRST 24 BYTES OF EID |

**Byte (hex)**

**Contents**

00–03 Record ID: C"CPP4"

04–06 0

07 Length of EID

08–1F First 24 bytes of EID

# CPRC entry for CPRC (Part 1)

**Entry: CPRC**

**VIT option:**

SSCP

**Event: CPRC macro (Part 1)**

**VIT processing module:**

ISTRACSC

**Control is returned to:**

Module invoking the INTRACE macro that caused the record to be produced

This trace record gives information about CPRC macroinstruction processing. It is written when a VTAM module issues a CPRC macroinstruction to set a nonzero sense code in an RUPE.

This event is treated as an exception condition and is always traced, whether the SSCP option is in effect or not.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 | 1 1 1 1 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 8 9 | A B C D E F |
| CPRC | I D | 0 | RUPE ADDRESS OR 0 | CPCB OPCODE IN THIS RUPE | SENSE CODE | ORIGIN NETWORK ADDRESS | DESTINATION NETWORK ADDRESS |

**Byte (hex)**

**Contents**

00–03 Record ID: C"CPRC"

04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05–07 0

08–0B Address of RUPE for this macro, or 0

**0C–0F** CPCB operation code for RUPE that indicates the original request for this
RUPE (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques
and Procedures*.)

**10–13** Sense code

**14–19** Network address at origin (RUPE)

**1A–1F** Network address at destination (RUPE)

## CPR2 entry for CPRC (Part 2)

**Entry: CPR2**
**VIT option:**
> SSCP

**Event:** CPRC macro (Part 2)
**VIT processing module:**
> ISTRACSC

This trace record is a continuation of the CPRC entry.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 | 0 | 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C | D | E | F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CPR2 | RETURN ADDRESS | ISSUER NAME | I N S T A N C E | R T N C D | 0 |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"CPR2"

**04–07** Address of the issuer of the CPRC macro

**08–0C** Module name field from the register save area for the module that issued
the CPRC macro

This field contains the abbreviated name (bytes 4, 5, 6, 7, 8) of the module
that issued the CPRC macro. If the issuing module does not have its own
save area, for example, modules ISTINCF1 and ISTPUCTI, this field
contains the name field from the save area of the caller of the module that
issued the CPRC macro.

**0D** Instance of the CPRC macro in the issuer

**0E** CPCBRC return code

> **Note:** It is not required that users of the product know the meaning of this
> internal VTAM return code. When required, the product support
> organization may use it to assist in internal flow diagnosis.

**0F–1F** 0

## CPR3 entry for CPRC (Part 3)

**Entry: CPR3**
**VIT option:**
> SSCP

**Event:** CPRC macro (Part 3)

VIT processing module:
     ISTRACSC

This trace record is a continuation of the CPRC entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C | D E F |
| CPR3 | 24 BYTES OF CPCBURC | 0 |

**Byte (hex)**
     **Contents**
00–03   Record ID: C"CPR3"
04–1C   First 25 bytes of the user request correlator (CPCBURC) entry
1D–1F   0

# CPWT entry for CPWAIT events (Part 1)

**Entry: CPWT**
**VIT option:**
     SSCP
**Event:** CPWAIT macro (Part 1)
**VIT processing module:**
     ISTRACSC
**Control is returned to:**
     ISTCPCPW

This trace record gives information about CPWAIT macroinstruction processing. It may help you determine why I/O is outstanding or why an SSCP request is failing.

This entry is eventually followed by a CPPT entry (to resume processing) or a CPPG entry (to purge the waiting task). Use the EID field to match corresponding entries.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CPWT | I D | 0 | T Y P E | 0 | LQAB ADDRESS | WREDATA IN WRE | WORK ELEMENT ADDRESS | CPCB OPCODE IN WORK EL OR 0 | SAVE AREA ID OR 0 | SENSE CODE OR 0 |

**Byte (hex)**
     **Contents**
00–03   Record ID: C"CPWT"
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05      0

| | |
|---|---|
| **06** | Type flags<br>• B'00000001'=PVI event<br>• B'00000000'=non-PVI event |
| **07** | 0 |
| **08–0B** | LQAB address for this CPWAIT macro |
| **0C–0F** | Data field in ISTWRE when CPWAIT is performed |
| **10–13** | Work element address |
| **14–17** | CPCB operation code that indicates the original request for this work element or 0 (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.) |
| **18–1B** | For PVI event, caller's save area ID or 0. For non-PVI event, save area ID |
| **1C–1F** | Sense code or 0 |

## CPW2 entry for CPWAIT (Part 2)

**Entry:   CPW2**
**VIT option:**
    SSCP
**Event:   CPWAIT macro (Part 2)**
**VIT processing module:**
    ISTRACSC

This trace record is a continuation of the CPWT trace record. If an RU is fewer than 24 bytes long, only the bytes in the actual RU will be moved to the corresponding fields in the trace records. The leftover bytes will be set to 0.

**Note:** This trace record is produced only for PVI events where a RUPE is available.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CPW2 | RETURN ADDRESS | FIRST 24 BYTES OF RU |

> **Byte (hex)**
> > **Contents**
> **00–03**   Record ID: C"CPW2"
> **04–07**   Return address
> **08–1F**   First 24 bytes of RU

## CPW3 entry for CPWAIT (Part 3)

**Entry:   CPW3**
**VIT option:**
    SSCP
**Event:   CPWAIT macro (Part 3)**
**VIT processing module:**
    ISTRACSC

This trace record is a continuation of the CPWAIT trace record.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C | D E F |
| CPW3 | RUPERC OR 0 | | 0 |

**Byte (hex)**

**Contents**

00–03 Record ID: C"CPW3"

04–1C RUPEURC or 0

1D–1F 0

## CPW4 entry for CPWAIT (Part 4)

**Entry: CPW4**

**VIT option:**

SSCP

**Event: CPWAIT macro (Part 4)**

**VIT processing module:**

ISTRACSC

This trace record is a continuation of the CPW3 trace record. If an event ID (EID) is fewer than 24 bytes long, only the bytes in the actual EID will be moved to the corresponding fields in the trace records. The leftover bytes will be set to 0.

**Note:** This trace record is produced only when the WRE points to an EID event.

| 0 0 0 0 | 0 0 0 | 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 | 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CPW4 | 0 | E I D L N | FIRST 24 BYTES OF EID |

**Byte (hex)**

**Contents**

00–03 Record ID: C"CPW4"

04–06 0

07 Length of EID

08–1F First 24 bytes of EID

## CPYB entry for COPY_DATA requests

**Entry: CPYB**

**VIT option:**

CSM

**Event: IVTCSM REQUEST=COPY_DATA**

**VIT processing module:**

ISTITCCS

**Control is returned to:**

IVTSMCCD

This trace record provides the status of an IVTCSM REQUEST=COPY_DATA macroinstruction.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D | 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CPYB | ID | 0 | FLAG | RECNUM | 0 | RTN COD | RSN COD | UTILRTN CALLER OR RETURN ADDRESS | NUMBER OF SOURCE BUFFERS | NUMBER OF TARGET BUFFERS | THREAD VALUE OR 0 |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"CPYB"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace record flag:

**B'0... ....'**
Indicates that this is the last record for this event.

**B'1... ....'**
Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07** Trace record number to correlate all the entries for this particular event

**08–0B** 0

**0C–0D**
Return code

**0E–0F** Reason code

**10–13** Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

**14–17** Number of buffers containing the data to be copied

**18–1B** Number of buffers to receive the copied data

**1C–1F** THREAD value if specified or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

# CPY2 entry for COPY_DATA requests

**Entry: CPY2**
**VIT option:**
CSM
**Event:** IVTCSM REQUEST=COPY_DATA
**VIT processing module:**
ISTITCCS

This trace record is a continuation of the CPYB trace record. It is an exception record and is generated only when the return code of the macroinstruction is not 0.

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CPY2 | 0 | F L A G | R E C N U M | LAST SOURCE BUFFER LIST ENTRY | LAST TARGET BUFFER LIST ENTRY | 0 |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"CPY2"

**04–05**  0

**06**  Trace record flag:

**B'0... ....'**
>   Indicates that this is the last record for this event.

**B'1... ....'**
>   Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07**  Trace record number to correlate all the entries for this particular event

**08–0B**  Last source buffer list entry successfully processed for error return codes

**0C–0F**  Last target buffer list entry successfully processed for error return codes

**10–1F**  0

# CPY3 entry for COPY_DATA requests

**Entry:  CPY3**

**VIT option:**
>   CSM

**Event:  IVTCSM REQUEST=COPY_DATA**

**VIT processing module:**
>   ISTITCCS

This trace record is a continuation of the CPYB trace record. It contains information about an entry in the source buffer list. There is one CPY3 trace record for each entry in the source buffer list.

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B C D E F 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| CPY3 | 0 | F L A G | R E C N U M | BUFFER TOKEN | BUFFER ALET | ADDRESS OF DATA BEING COPIED | SIZE OF DATA BEING COPIED |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"CPY3"

**04–05**  0

**06**  Trace record flag:

B'0... ....'
>    Indicates that this is the last record for this event.

B'1... ....'
>    Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

| | |
|---|---|
| 07 | Trace record number to correlate all the entries for this particular event |
| 08–13 | Buffer token |
| 14–17 | ALET |
| 18–1B | Address of data being copied |
| 1C–1F | Size of data being copied |

## CPY4 entry for COPY_DATA requests

**Entry:** CPY4
**VIT option:**
>    CSM
**Event:** IVTCSM REQUEST=COPY_DATA
**VIT processing module:**
>    ISTITCCS

This trace record is a continuation of the CPYB trace record. It contains information about an entry in the target buffer list. There is one CPY4 trace record for each entry in the target buffer list.

| 00 01 02 03 | 04 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|
| CPY4 | 0 | F L A G | R E C N U M | BUFFER TOKEN | BUFFER ALET | STORAGE ADDRESS WHERE DATA IS TO BE COPIED | STORAGE LENGTH WHERE DATA IS TO BE COPIED |

**Byte (hex)**
>    **Contents**

| | |
|---|---|
| 00–03 | Record ID: C"CPY4" |
| 04–05 | 0 |
| 06 | Trace record flag: |

B'0... ....'
>    Indicates that this is the last record for this event.

B'1... ....'
>    Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

| | |
|---|---|
| 07 | Trace record number to correlate all the entries for this particular event |
| 08–13 | Buffer token |
| 14–17 | Access List Entry Token (ALET) |
| 18–1B | Address of the storage where data is to be copied |
| 1C–1F | Length of the storage where data is to be copied |

# CRx entry for ADD, DELETE, or FIND control block

**Entry:  CRA, CRD, or CRF**
**VIT option:**
     SSCP
**Event:**  CRADD, CRDEL, CRFIND
**VIT processing module:**
     ISTRACSC

The correlate search control block (CR) trace record contains information about a control block add, delete, or find operation in the VTAM control point (CP).

| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 4 5 6 7 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| CRA CRD CRF | T Y P E | I D | R C | MODIFIER LIST, SEARCH NUMBER, OR 0 | RETURN ADDR | PCID OR 0 | CORCB ADDR OR 0 |

**Byte (hex)**

**00–02**  Record ID:
   **C"CRA" (CRADD):**
        Add control block
   **C"CRD" (CRDEL):**
        Delete control block
   **C"CRF" (CRFIND):**
        Find control block

**03**  Type code
   **0**     Procedure-correlation identifier (PCID) only (6–F contain zeros)
   **1**     PCID and modifier list (6–F contain the modifier list)
   **2**     PCID and search number (6–7 contain the search number, 8–F contain zeros)

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  Return code if specified; some FIND calls to this macro do not include a return code but use CORCB ADDR as a return code.

**06–0F**

• If type code = 0, this field is 0.

• If type code = 1, this field represents the PCID modifier list and is padded on the right with zeros. If search number correlation is used, this field may be 0.

• If type code = 2, this field represents the search number and is padded on the right with zeros.

**10–13**  Calling module's return address
**14–1B**  PCID of the search being correlated
**1C–1F**  Address of an ISTCORCB; bytes 4–7 of the ISTCORCB contain the address of the control block being correlated (LCB, SITCB). This field is 0 if the entry type is "CRF" and the control block was not found.

**Notes:**

1. A CRF entry is usually found soon after a DSP entry for LUSS or DSVC. If the CORCB ADDR is 0, the FIND failed.

2. Search number correlation is only used with DS.

---

# CSx entry for resource state change

**Entry:   CSC, CSD, or CSB**
**VIT option:**
   SSCP
**Event:  Change in resource state**
**VIT processing module:**
   ISTRACSC
**Control is returned to:**
   The module that issued the INTRACE macroinstruction

This trace record is written when the current state or desired state, or both, of a resource for which tracing has been requested changes. If the network ID of the resource being traced differs from the network ID of the host, the trace record CSC2, CSD2, or CSB2 will follow this entry.

| 00 01 02 | 03 | 04 05 | 06 07 | 08 09 0A 0B 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|
| CSC CSD CSB | RPRENTRY | RPRCURST | RPRDEST | RESOURCE NAME | RETURN ADDRESS | RESOURCE ADDRESS | MODULE NAME | RPH ADDRESS |

**Byte (hex)**
   **Contents**
**00–02**   Record ID:
   • C"CSC" for current state change
   • C"CSD" for desired state change
   • C"CSB" for both current and desired state change
**03**   Type of resource
**04–05**   Resource's current state
**06–07**   Resource's desired state
**08–0F**   Resource name

   **Note:** If the resource RDTE is not available, RDTEPTR is 0, bytes 04–0F will be 0.
**10–13**   Return address of the module that changed the resource's state
**14–17**   Address RDTE
**18–1B**   Name of the module that changed the resource's state
**1C–1F**   Request parameter header (RPH) address

## CSC2, CSD2, or CSB2 entry for resource state change
**Entry:   CSC2, CSD2, or CSB2**
**VIT option:**
   SSCP
**Event:  Change in resource state of resource with network ID different from host's network ID**

**VIT processing module:**
ISTRACSC

This trace record is a continuation of the CSx entry and is written when the network ID of the resource being traced differs from the network ID of the host.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| CSC2 CSD2 CSB2 | NET ID | 0 |

**Byte (hex)**
   **Contents**
**00–03**   Record ID:
   - C"CSC2" for current state change
   - C"CSD2" for desired state change
   - C"CSB2" for both current and desired state change
**04–0B**   Network ID of the resource being traced
**0C–1F**   0

---

# DAPT entry for HPR out-of-sequence or received segments queue DAPTR

**Entry:  DAPT**
**VIT option:**
   HPR
**Event:  DAPTR alteration**
**VIT processing module:**
   ISTITCHP

This trace record shows most of the DAPTR fields. The DAPTR is used to chain inbound HPR data that arrives out of order or in order but is segmented into pieces. The DAPTR resides either on the RPN_OutOfSeq_Msg_Q or RPN_RCV_Segments_DaPtr queue of the RPNCB that is currently dispatched. The record shows the updated contents of the DAPTR.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| DAPT | A S I D | 0 | N L P C N T | D A P F L A G S | First Head TSCB Address | Last Head TSCB Address | Begin Seq Number | End Seq Number | RPNCB Address | DAPTR Address |

**Byte (hex)**
   **Contents**
**00–03**   Record ID: C"DAPT"

| | |
|---|---|
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | NLP count (X'FF' if 255 or greater) |
| **07** | DAP_Flags: |

| Bit | Meaning |
|---|---|
| **1... ....** | DAP_End_Msg |
| **.1.. ....** | DAP_Beg_Msg |
| **..1. ....** | DAP_Last_Msg |
| **...1 ....** | DAP_COB_Ind |
| **.... xxxx** | 0 |

| | |
|---|---|
| **08–0B** | DAP_First_Head_TSCB |
| **0C–0F** | DAP_Last_Head_TSCB |
| **10–13** | DAP_Begin_Seq |
| **14–17** | DAP_End_Seq |
| **18–1B** | RPNCB address |
| **1C–1F** | DAPTR address |

---

# DBx entry for DBDELETE, DBQUERY, and DBUPDATE

**Entry:  DBD, DBQ, DBU**
**VIT option:**
       SSCP
**Event:  DBDELETE, DBQUERY, DBUPDATE**
**VIT processing module:**
       ISTRACSC

This traces an operation on the APPN directory database.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DBD DBQ DBU | | | 0 | I D | R C | F L G S | | RESOURCE IDENTIFICATION | | | | | | | | RETURN ADDRESS | | | | DECB ADDRESS | | | | NETID OF RESOURCE | | | | | | | |

**Byte (hex)**
       **Contents**

| | |
|---|---|
| **00–02** | Record ID: C"DBD", C"DBQ", C"DBU" |
| **03** | 0 |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | Return code (RC) |
| **06–07** | Database information flags |

| Bit | Meaning |
|---|---|
| **1... ....** | Registered entry-type indicator |

.1.. ....
: Cache or dynamic entry-type indicator

..1. ....
: Suggestion or defined entry-type indicator

...1 ....
: Reserved for entry-type expansion

.... 1...
: The VIT error was caused by an error in the cache data processing

.... .1..
: The VIT error was caused by an error while loading the directory from storage

.... ..1.
: The returned network ID on DBQUERY differs from the one queried

.... ...x
: Not used

1... ....
: Directed failed indicator copied from directory entry

.1.. ....
: Negative cache indicator copied from directory entry

..1. ....
: Subarea LU indicator copied from directory entry

...1 ....
: Surrogate owner indicator copied from directory entry

.... 1...
: Dynamic subarea destination LU indicator copied from directory entry

.... .1..
: Wildcard LU indicator copied from directory entry

.... ..1.
: Nonnative LU indicator copied from directory entry

.... ...1
: Generic name indicator copied from directory entry

**08–0F**  Resource identification
**10–13**  Caller's return address
**14–17**  Address of directory entry
**18–1F**  Network identification of resource

# DCON entry for discarded container

**Entry: DCON**
**VIT option:**
: PIU
**Event:** Discard Container
**VIT processing module:**
: ISTRACOT
**Control is returned to:**
: ISTTSCUA

This trace record is written when the VTAM Enterprise Extender utility discard routine, ISTTSCUA, disposes of a container. The reason code can be used to explain the reason for the discard.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 | A B | C D E F |
| DCON | ID | 0 | CON-TAINER ADDRESS | REASON CODE | TARGET IPv4 ADDRESS OR ZEROS | PLIST ADDRESS | S O U R C E | T A R G E T | RPH ADDRESS |

**Byte (hex)**

> **Contents**

**00–03** Record ID: C'DCON'

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** 0

**08–0B** Container address

**0C–0F** Reason code

> **Code** **Meaning**
>
> **X'0000'**
>> VTAM is not able to contact UDP.
>
> **X'0002'**
>> Data is sent successfully, but fragmented.
>
> **X'0004'**
>> Temporary error. Retry.
>
> **X'0008'**
>> Error. Local IP address is not valid.
>
> **X'000C'**
>> Error. Local IP address is not a valid VIPA address.
>
> **X'0010'**
>> Error. Port cannot be reserved.
>
> **X'0014'**
>> Error. Parameter is not valid.
>
> **X'0018'**
>> Error. State is not valid.
>
> **X'001C'**
>> Error. Destination for datagram is unreachable.
>
> **X'0020'**
>> Error. VTAM is not authorized.
>
> **X'0024'**
>> Error. Storage unavailable.
>
> **X'0028'**
>> Error. Data exceeds maximum.
>
> **X'002C'**
>> Error. Stack is not valid.
>
> **X'0040'**
>> Permanent error.

**10–13** Target IPv4 address or zeros. If zeros, target IPv6 address is reported in DCO2 trace record.

**14–17** Parameter list address

**18–19** Source port number

**1A–1B** Target port number

**1C–1F** Request parameter header (RPH) address

## DCO2 entry for discarded container (Part 2)

**Entry: DCO2**
**VIT option:**
> PIU

**Event:** Discard container
**VIT processing module:**
> ISTRACOT

This trace record is a continuation of the DCON entry. This entry will be present only when the discarded container contains an IPv6 address.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|---|
| DCO2 | 0 | IPv6 ADDRESS |

**Byte (hex)**
> **Contents**

00–03   Record ID: C'DCO2'
04–0F   0
10–1F   IPv6 Address

## DEVx entry for MPNCB device counter update

**Entry: DEVD or DEVI**
**VIT option:**
> CIA

**Event:** MPNCB Active device counter update
**VIT processing module:**
> ISTRACCI

**Control is returned to:**
> Modules invoking the INTRACE macro that caused the record to be produced.

This trace record is written when a module is about to update an active device counter in MPNCB. It records the current Read and Write device counters before the change takes place.

- DEVD is generated for a counter decrementation.
- DEVI is generated for a counter incrementation.

Refer to *z/OS Communications Server: SNA Data Areas Volume 1* for a description of the NCB fields.

For record types with suffix I,X, or T the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DEVD DEVI | | | | ID | 0 | | | MODULE NAME | | | | CPNCB ADDRESS | | | | MPN-ARTPH (ACTIVE READ CPNCBs) | | | | 0 | | | | MNP-AWTPH (ACTIVE WRITE CPNCBs) | | | | RPH POINTER | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID:

C"DEVD" for Device counter decrementation

C"DEVI" for Device counter incrementation

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07**  0

**08–0B**  The module name that invokes the INTRACE macro.

**0C–0F**  The CPNCB address of the device that is about to be excluded (DEVD) or included (DEVI) from the MPNCB current active device counter.

**10–13**  Active Read device counter (MPNARTPH)

**14–17**  0

**18–1B**  Active Write device counter (MPNAWTPH)

**1C–1F**  RPH address

# DLT entry for directory services locate (Part 1)

**Entry:  DLT**

**VIT option:**

SSCP

**Event:**  Sending out a Locate Search

**VIT processing module:**

ISTRACSC

**Control is returned to:**

ISTDRSDL

This trace record is written when VTAM's directory services component forwards a locate search to some adjacent node.

**Note:**  This trace record has been reformatted because of an increase in the size of the task vector field.

| 0 0 0 0 0 | 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 | 1 | 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 4 | 5 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B | C D E F |
| DLT | 0 | I D | 0 | TASK VECTOR | RETURN ADDRESS | N O D E R O L E | D S M E R C | T E R M C O N D | LCB ADDRESS | SENSE CODE |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"DLT"

**03** 0

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** 0

**08–0F** The task vector bits indicate the possible tasks for this locate search. The hex values listed below are in the task vector field of this entry and the task vector results field of the DLT2 entry.

**08**

| Vector | Description |
|---|---|
| X'80' | Null task |
| X'40' | Directory services management exit |
| X'20' | Directory services database query |
| X'10' | Topology and routing services database query |
| X'08' | Forward to network node server |
| X'04' | One hop if directory services database is found |
| X'02' | One hop if control vector X'0E' is received for request |
| X'01' | Nonverify attempt |

**09**

| Vector | Description |
|---|---|
| X'80' | Directed due to network node destination LU hierarchy received on a search request |
| X'40' | Directed if directory services database is found |
| X'20' | Directed if the topology and routing services valid route selection CV is returned |
| X'10' | Resource Discovery Search |
| X'08' | Directed to a directory server |
| X'04' | Directed to a higher function directory server |
| X'02' | Directed to a directory server retry |
| X'01' | Directed to a gateway node |

**0A**

| Vector | Description |
|---|---|
| X'80' | Sequential directed search to alternate directory servers |
| X'40' | Sequential directed search to interchange nodes |
| X'20' | Subarea system resolution table (SRT) cache search |
| X'10' | Subarea search after a positive cache search |
| X'08' | Subarea search after a positive directory services database query |
| X'04' | Subarea search after a negative or no cache search |

|        |        |
|--------|--------|
| **X'02'** | Domain broadcast search |
| **X'01'** | Originate network broadcast search |

**0B**

| Vector | Description |
|--------|-------------|
| **X'80'** | Forward network broadcast not originated by this node |
| **X'40'** | One-hop search request due to end node destination LU hierarchy received on a search request |
| **X'20'** | A cross-subnetwork directed search due to information received on the original request |
| **X'10'** | A cross-network directed search due to information found in the directory services database |
| **X'08'** | A directed search due to information found in the topology and routing services database |
| **X'04'** | Sequential directed search with the intent of finding the resource cross-subnetwork |
| **X'02'** | Generic cache search |
| **X'01'** | A directed search due to a SEARCH_RPY interprocess signal following a positive CACHE_SEARCH_RPY interprocess signal |

**0C**

| Vector | Description |
|--------|-------------|
| **X'80'** | A directed search due to a SEARCH_RPY interprocess signal following a positive directory services database query |
| **X'40'** | A subarea search due to a SESS_INIT_INFO_RPY interprocess signal |
| **X'20'** | Database query after an RDS |
| **X'10'** | Sequential directed search to other network nodes in the generic resource configuration |
| **X'08'** | Final subarea search after resource not found in APPN with SSEARCH = APPNFRST |

**0D–0E** 0

**0F**

| Vector | Description |
|--------|-------------|
| **X'01'** | Post processing |

**10–13** Caller's return address

**14–15** The node role for this search. More than one bit can be on.

| Code | Description |
|------|-------------|
| **X'80'** | CP originating LU |
| **X'40'** | CP destination LU |
| **X'20'** | NN originating LU |
| **X'10'** | NN destination LU |
| **X'08'** | Owning directory server |
| **X'04'** | Alternate directory server |
| **X'02'** | Intermediate network server |
| **X'01'** | Intermediate network node directed |

| Position | Description |
|----------|-------------|
| **X'80'** | Intermediate network node broadcast |

**16** Return code from the directory services management exit

**17** Terminating condition indicates why the search ended

| Code | Description |
|------|-------------|
| **X'00'** | Processing can continue. |
| **X'04'** | Positive reply can be returned to parent. |

| | |
|---|---|
| **X'08'** | Gateway reply has been received. |
| **X'0C'** | Directory server reply has been received. |
| **X'10'** | An error was detected by a task called from the sequencer. |
| **X'14'** | Cleanup is pending after all replies are received from the search phase (CP session outage). |
| **X'18'** | A directed search was performed due to wildcard information, and a wildcard was returned. No further searching will be done. |
| **X'20'** | An alternate directory server had an unknown (neg cache) entry. |
| **X'24'** | Directory services management exit routine specified no search for this request. |
| **X'28'** | An error was detected during generic cache search task. |
| **X'32'** | This search kicked off an RDS which returned a negative reply. No more searching should be done for this search. |
| **X'34'** | IOPURGE occurred during verification after RDS found resource. |
| **X'36'** | Directed search failed after RDS found resource. |
| **18–1B** | Address of the locate control block for this search. |
| **1C–1F** | Sense code. |

## DLT2 entry for directory services locate (Part 2)

**Entry: DLT2**
**VIT option:**
　　SSCP
**Event:** Sending out a Locate Search
**VIT processing module:**
　　ISTRACSC

The DLT2 trace record is a continuation of the DLT trace record.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| DLT2 | 0 | TASK VECTOR RESULTS | 0 |

**Byte (hex)**
　　**Contents**
**00–03** Record ID: C"DLT2"
**04–07** 0
**08–0F** Task Vector Results

　　Use the flags in the task vector results field to determine whether the task indicated in bytes 08—0F in the DLT entry were invoked.
　　**B'0'** Task was not invoked.
　　**B'1'** Task was invoked.
**10–1F** 0

## DRPx entry for DLURRTP macroinstruction invocation (Part 1)

**Entry: DRPA, DRPC, or DRPD**

**VIT option:**
> HPR

**Event:** Invocation of DLURRTP macroinstruction

**VIT processing module:**
> ISTITCHR

**Control is returned to:**
> Module invoking the DLURRTP macroinstruction

This trace record is written when the DLURRTP macroinstruction is issued. It indicates the function being performed, the return code for that function, and the addresses of the parameters involved.

| 0 0 0 0 | 0 0 | 0 0 | 0 | 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 7 | 8 | 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| DRPA DRPC DRPD | I D  0 | U S E C O U N T | R C | F L A G S  0 | DLUR RTP ADDRESS | RETURN ADDRESS | TCID | RPH ADDRESS |

**Byte (hex)**

> **Contents**

**00–03** Record ID:
- C"DRPA" for DLURRTP ADD
- C"DRPC" for DLURRTP CHG
- C"DRPD" for DLURRTP DEL

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06–07** Use count is the number of sessions associated with this DLUR RTP entry.

**08** Return code:

> **00** Success

> **04** Not found

> **08** Insufficient storage

> **16** Operation aborted due to major error

**09** Flags (as defined in DLURRTP control block)

> **Bit** **Meaning**
> **1... ....**
> > NetView notified of path switch for this RTP connection.
> **.1.. ....**
> > RTP connection is in use by at least one session (has meaning only during NLDM start processing).
> **..1. ....**
> > UNCOND specified (applicable to DLURRTP DEL only).
> **...1 ....**
> > TCID or CQF information input to macroinstruction is not in a valid format.

> **.... 1...**
>> RTP RSCV could not be saved.
>
> **.... .xxx**
>> Available.

**0A–0B**  0

**0C–0F**  Address of DLUR RTP control block

**10–13**  Address of issuer of the DLURRTP macroinstruction

**14–1B**  Data field of the transport connection identifier (TCID) control vector (X'4B')

**1C–1F**  Request parameter header (RPH) address

# DRP2 entry for DLURRTP macroinstruction invocation (Part 2)

**Entry:  DRP2**

**VIT option:**
> HPR

**Event:**  Invocation of DLURRTP macroinstruction

**VIT processing module:**
> ISTITCHR

This trace record is a continuation of the DRP entry. It contains the four data fields from the network address control vector, left-justified in fields of the maximum length allowed.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
|---|---|---|---|---|
| DRP2 | NETWORK ID | CP NAME | NCE ID | NCE INSTANCE |

**Byte (hex)**
> **Contents**

**00–03**  Record ID: C"DRP2"

**04–0B**  Network ID of the DLUR

**0C–13**  CP name of the DLUR

**14–1B**  NCE identifier

**1C–1F**  NCE instance identifier

# DRP3 entry for DLURRTP macroinstruction invocation (Part 3)

**Entry:  DRP3**

**VIT option:**
> HPR

**Event:**  Invocation of DLURRTP macroinstruction

**VIT processing module:**
> ISTITCHR

This trace record is a continuation of the DRPA and DRPC entries. It is not written for the DRPD entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| DRP3 | 28 BYTES OF RSCV |

**Byte (hex)**
> **Contents**
**00–03** Record ID: C"DRP3"
**04–1F** 28 bytes of RSCV representing the RTP connection

# DSCx entry for discarded TSCBs or TIPAC (Part 1)

**Entry: DSCD or DSCO**
**VIT option:**
> PIU

**Event:** Discarded TSCB or TIPAC (Part 1)
**VIT processing module:**
> ISTRACOT

**Control is returned to:**
> ISTTSCUD

This trace record is written when VTAM's TSC utility discard routine, ISTTSCUD, disposes of transmission subsystem control blocks (TSCBs) or transport interface parameter access containers (TIPACs) because of an error condition. The reason code and module identifier can be used to explain the reason for the discard. This entry is treated as an exception condition and is always traced if the VIT is active, regardless of the VIT options specified. A DSC2 entry follows a DSCO or DSCD entry. The DSC2 entry includes the ID of the module that is discarding the storage.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| DSCD DSCO | I D | REASN CODE | TSCB ADDRESS | 20 BYTES OF TSCBs OR TIPACs |

**Byte (hex)**
> **Contents**
**00–03** Record ID:
> - C"DSCD" for discarding from procedural modules
> - C"DSCO" for discarding from object-oriented methods

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07** Discard reason code
**08–0B** Address of TSCB
**0C–1F** First 20 characters of the TSCB or TIPAC.

## DSC2 entry for discarded TSCB or TIPAC (Part 2)

**Entry:  DSC2**
**VIT option:**
>       PIU
**Event:**  Discarded TSCB or TIPAC (Part 2)
**VIT processing module:**
>       ISTRACOT

This trace record is a continuation of the DSCx entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| DSC2 | 24 MORE CHARACTERS OF THE TSCB OR TIPAC | MODULE ID |

**Byte (hex)**
>       **Contents**
**00–03**  Record ID: C"DSC2"
**04–1B**  24 more characters of the TSCB or TIPAC
**1C–1F**  ID of the module that is discarding the TSCB or TIPAC

# DSP entry for PAB dispatch

**Entry:  DSP**
**VIT option:**
>       PSS
**Event:**  PAB dispatch
**VIT processing module:**
>       ISTRACPS
**Control is returned to:**
>       ISTAPCPD

The DSP entry marks the beginning of a PAB dispatch, which is the VTAM major unit of work. This unit exists until terminated by an EXIT entry. While processing, it might wait for other events to complete (WAIT). It resumes processing with a RESM entry. The unit might be interrupted by a higher-priority task and continue when that task is finished.

Each PAB is identified by a unique index number. VTAM uses this number to find the called module and then creates a VIT entry for it.

To correlate all events associated with this unit, look for the request parameter header (RPH) address, which is in many trace records, and match it to the request parameter header (RPH) address in the DSP trace record. This unit has exclusive use of that RPH until EXIT.

| 0 0 0 | 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| DSP | 0 / W E Q | I D | C B I D | F L A G S | PST ADDRESS | PAB ADDRESS | LAST WORK ELEMENT ADDRESS | CURRENT WORK ELEMENT ADDRESS | MODULE NAME OR DVT ADDRESS | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"DSP"

**03** 0, or, for very extended PABs, PAB work element queue level dispatched

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID of work element (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)

**06** PAB flag field (PABFLAGS)

**Bit** **Meaning**

**1... ....**
PAB is unconditionally scheduled.

**.1.. ....**
PAB closedown is in progress.

**..1. ....**
PAB is synchronous.

**...1 ....**
PAB extension is present.

**.... 1...**
Do not dequeue work element.

**.... .1..**
Do not detach the request parameter header (RPH).

**.... ..1.**
Indicates a very extended PAB.

**.... ...1**
Indicates a slightly extended PAB.

**07** PAB flag field (PABFLGS1)

**Bit** **Meaning**

**1... ....**
Switch the PST address of the major control block for this PAB to the new PST address contained in DYPNWPST.

**.1.. ....**
This PAB has a data space extension.

**..1. ....**
This PAB's major control block is an FMCB.

**...1 ....**
PAB can be referenced in PSW disable mode.

**.... 1...**
PAB is persistent.

**.... .1..**
APSTERM/APSINIT FMCB during PAB dispatch.

**.... ..xx**
Reserved.

**08–0B** PST address

**0C–0F** PAB address

**10–13** Address of work element most recently queued to the PAB
**14–17** Address of work element currently being dispatched
**18–1B** Module name abbreviation (usually bytes 4, 5, 7, and 8 of the module name) or PAB DVT address (high order bit of X'18' = 0). For an explanation of the module-naming convention, see "Module names in internal trace records" on page 21. (The module name might be unavailable if the PAB being scheduled is associated with an address space different from the current one.)
**1C–1F** Request parameter header (RPH) address

# DTSK entry for detach a subtask

**Entry: DTSK**
**VIT option:**
> PSS

**Event:** Detach a subtask.
**VIT processing module:**
> ISTRACPS

**Control is returned to:**
> The module that issued the DETACH

This trace record is written when a VTAM module detaches a VTAM subtask.

Some subtasks are detached without generating this entry.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DTSK | | | ID | RC | | 0 | | TASK NAME OR 0 | | | | | | | | TCB ADDR | | | | 0 | | | | ISSUER ADDRESS | | | | 0 | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"DTSK"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Return code from the detach macro (ITKRC)
**06–07** 0
**08–0F** Name of the subtask being detached, or 0
**10–13** TCB address of the subtask being detached
**14–17** 0
**18–1B** Address of the issuer of the DETACH macro
**1C–1F** 0

# ENFx entry for ENF exit

**Entry: ENFF, ENFN, ENFP, or ENFR**
**VIT option:**
> CIA

**Event:** ENF exit for APPN host-to-host channel dynamics
**VIT processing module:**
> ISTRACCI

**Control is returned to:**
> ISTTSCDY

This trace record is written when MVS schedules the VTAM ENF exit for APPN host-to-host channel dynamics for a subchannel device.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| ENFF ENFN ENFP ENFR | | | | ID | RESERVED | 0 | | CUA | | | | SCL ADDRESS | | | | SCL ENF | 0 | SCL STATES | | 0 | | | | | | | | RPH ADDR | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID:
- C"ENFF" for offline
- C"ENFN" for online
- C"ENFP" for pending offline
- C"ENFR" for reaccessible

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Reserved

**06–07** 0

**08–0B** CUA from UCBCHAN or DACH_IORA_DEVN

**0C–0F** Address of SCL

**10** SCL ENF flags

**11** 0

**12–13** SCL states:
- **12** SCL channel state
- **13** SCL_SYS_state

**14–1B** 0

**1C–1F** Request parameter header (RPH) address

# ENR entry for APPN resource registration processing

**Entry: ENR**
**VIT option:**
SSCP
**Event:** An APPN resource registration function is being processed.
**VIT processing module:**
ISTRACSC

This trace record is written by end nodes when Configuration Services processes an APPN registration. The purpose of the ENR trace record is to record the update and delete FSM states before and after registration. This entry also provides information on which process invoked the registration function and what processing was done.

**Note:** This trace record is written at the conclusion of the registration process.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENR | | | 0 | I D | 0 | | R C | E U F S M | C U F S M | E D F S M | C D F S M | E V E N T | 0 | | F L A G S | MODULE ID | | | | PROCESS ID | | | | INVOKER ID | | | | RPH | | | |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"ENR"

**03** 0

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–06** 0

**07** RC is the return code on exit from ISTCSCRE.

**08** EUFSM is the UPDATE FSM state upon entry to ISTCSCRE. The FSM states are defined within ISTCSCRE.

**09** CUSFM is the changed UPDATE FSM's new value. The changed value may be the same as the entry value.

**0A** EDFSM is the DELETE FSM state upon entry to ISTCSCRE. The FSM states are defined within ISTCSCRF.

**0B** CDFSM is the changed DELETE FSM's value after processing by ISTCSCRF. The changed value may be the same as the entry value.

**0C** EVENT is the registration event code that is being processed. These codes are defined in ISTREVNT.

**0D–0E** 0

**0F** Flags

**Bit**

**11.. ....** Server Availability Status

    **00** Not available

    **01** Available

    **10** Pending

**..11 ....** Request type

    **01** Network node server request

    **10** Central Resource Registration request

    **11** Delete request

**.... 1...** Reply pending because server is not available

**.... .1..** Processing USERVAR

**10–13** Module issuing the INTRACE macro

**14–17** Indicates which processing routine was invoked

**18–1B** Indicates which module invoked the registration process

**1C–1F** Request parameter header (RPH) address

# ERPx entry for error recovery procedures

**Entry:** ERPI, ERPL, ERPT, or ERPX

**VIT option:**
    CIO

**Event:** Error recovery

**VIT processing module:**
    ISTRACCI

**Control is returned to:**
ISTZBM0J for LDNCB, or ISTZBM0K for ICNCB and RWNCB

This trace record is written during error recovery for channel I/O.

- ERPI is generated for communication controllers and local SNA cluster controllers.
- ERPL is generated for local non-SNA cluster controllers.
- ERPT is generated for IP over channel data link control connections.
- ERPX is generated for channel-to-channel-attached hosts and channel-attached IBM 3172 Interconnect Nways Controllers.

For record types with suffix I, X, or T, the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| ERPI ERPL ERPX ERPT | | | | ID | STATE | 0 | 0 | DEVICE | | | | NCB ADDRESS | | | | FLAG BYTES | | | | FLAG | CODE | SENSE OR 0 | | CSW | | | | | | | |

**Byte (hex)**

**Contents**

00–03  Record ID:
  C"ERPI" for ICNCB
  C"ERPL" for LDNCB
  C"ERPT" for RWNCB
  C"ERPX" for XCNCB

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05  For ERPI, ERPL, and ERPT, link station state (NCBLNKST). For ERPX, station state (XCNSSFSM)

06  0

07  0

08–0B  Channel device name in EBCDIC (either a device address or device number)

0C–0F  NCB address

10–13  Flag bytes (NCBFLAGS)

14  Flag byte as follows:

**Bit    Meaning**

..1. ....
  Exception condition occurred (IOSEX flag is on)

...1 ....
  Error routine is in control (IOSERR flag is on)

15  I/O completion code (IOSCOD)

16–17  For ERPI, ERPL and ERPT, sense data (IOSSNS). For ERPX, 0.

18–1F  Channel status word from IOSB

# ESC entry for the TPESC macro

**Entry: ESC**
**VIT option:**
    ESC
**Event: TPESC macro**
**VIT processing module:**
    ISTRACOT
**Control is returned to:**
    ISTESC01

This trace record identifies the address of the next routine to get control in a destination vector table (DVT). The PAB address identifies the process. Using the request parameter header (RPH) address, you can correlate this trace record to the DSP entry to determine when the process was dispatched.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|
| ESC | I D | 0 | P A B O F | DSP ADDRESS | PAB ADDRESS | RETURN ADDRESS | RPH WORK ELEMENT | MODULE NAME OR DVT ADDRESS | RPH ADDRESS |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"ESC"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06–07** PAB offset
**08–0B** Dispatch address from DVT
**0C–0F** PAB address
**10–13** Address of the issuer of the TPESC macro
**14–17** RPH work element
**18–1B** One of the following:

- Module name abbreviation (bytes 4, 5, 7, and 8 of the name of the next module to get control)
- PAB DVT address (high order bit of X'18' = 0)

**1C–1F** Request parameter header (RPH) address

# ETSK entry for exit a subtask

**Entry: ETSK**
**VIT option:**
    PSS
**Event: Exit a subtask**
**VIT processing module:**
    ISTRACPS
**Control is returned to:**
    The subtask that is terminating

This trace record is written when a VTAM subtask is about to exit and return to the operating system supervisor.

This is not fully implemented for all subtask events in VTAM. There are a number of subtasks that terminate without this entry being generated.

| 0 0 | 0 0 | 0 | 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|0 1 2 3|4|5 6 7|8 9 A B C D E F|0 1 2 3|4 5 6 7 8 9 A B|C D E F|
| ETSK | I D | 0 | TASK NAME | TCB ADDR | 0 | REG 15 |

**Byte (hex)**

   **Contents**

**00–03**   Record ID: C"ETSK"

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07**   0

**08–0F**   Name of the subtask about to end execution

**10–13**   TCB address of the current subtask

**14–1B**   0

**1C–1F**   Contents of register 15

# EXIT entry for the TPEXIT macro

**Entry:   EXIT**

**VIT option:**

   PSS

**Event:   TPEXIT macro**

**VIT processing module:**

   ISTRACPS

**Control is returned to:**

   ISTAPCTX

This trace record identifies a VTAM process (PAB) that has finished executing. The PAB is rescheduled if more work elements are waiting to be processed.

In most cases, the request parameter header (RPH), whose address is in location 1C–1F, is freed or reused after this entry. One exception is the SMS dynamic expansion DYPAB (ATCPXPAB in module ISTORFPX), which has a dedicated RPH that is not used for anything else.

This entry is the complement of DSP. An EXIT should eventually follow every DSP. Use the RPH address to correlate the entries.

| 0 0 | 0 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|0 1 2 3|4|5|6 7|8 9 A B|C D E F|0 1 2 3|4 5 6 7|8 9 A B|C D E F|
| EXIT | I D | O P T S | P A B O F | PST ADDRESS | PAB ADDRESS | RETURN ADDRESS | WORK EL Q OR NEXT DSP Q LVL | MODULE NAME OR DVT ADDRESS | RPH ADDRESS |

**Byte (hex)**

   **Contents**

| | |
|---|---|
| **00–03** | Record ID: C"EXIT" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | TPEXIT options |
| **06–07** | PAB offset, or 0 if no PAB |

      **Note:** There is no PAB if the running process has already freed the major control block that contains the PAB.

| | |
|---|---|
| **08–0B** | PST address |
| **0C–0F** | PAB address, or 0 if no PAB |

      **Note:** There is no PAB if the running process has already freed the major control block that contains the PAB.

| | |
|---|---|
| **10–13** | Address of the issuer of the TPEXIT macro |
| **14–17** | One of the following: |

- PAB work element queue
- Next dispatchable queue level for a very extended PAB (see corresponding DSP entry to find the PABFLAGS to determine the PAB type)
- 0, if no PAB

      **Note:** There is no PAB if the running process has already freed the major control block that contains the PAB.

| | |
|---|---|
| **18–1B** | Module name abbreviation (usually bytes 4, 5, 7, and 8 of the module name) or PAB DVT address (high order bit of X'18' = 0). For an explanation of the module naming convention, see "Module names in internal trace records" on page 21. (The module name might be unavailable if the PAB being scheduled is associated with an address space different from the current one.) |
| **1C–1F** | Request parameter header (RPH) address |

**Note:** There is no PAB if the running process has already freed the major control block that contains the PAB.

## EXPN entry for buffer pool expansion

**Entry:** EXPN
**VIT option:**
      SMS
**Event:** Buffer pool expansion
**VIT processing module:**
      ISTRACSM
**Control is returned to:**
      ISTORAPX

This trace record is generated whenever VTAM needs to expand a buffer pool. Buffer pool expansion may occur because of a scheduled request.

| 00 01 02 03 | 04 | 05 06 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 | 16 | 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| EXPN | ID | 0 | BPCB ADDRESS | PXB ADDRESS | EXTENT ADDRESS | PAGES | CODE | FLAGS | TOTAL NUMBER OF BUFFERS | AVAIL NUMBER OF BUFFERS |

**Byte (hex)**

**Contents**

00–03  Record ID: C"EXPN"

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05–07  0

08–0B  Address of buffer pool control block (BPCB) for which expansion occurred

0C–0F  Address of pool expansion block (PXB) that is expanded

10–13  Extent address (PXBSTADR)

14–15  Number of pages for an expansion in this pool. Number of pages for an expansion = (BPCBEXLN divided by 4096).

16  Expansion failure code if expansion failed

> **Note:** VTAM issues codes 4 through 8 when a failure occurs during a deferred expansion.

> **4**   Not enough CSA storage is available for the expansion.

> **5**   VTAM cannot fix pages in storage due to insufficient page frames or some other page locking problem.

> **7**   Storage unavailable. VTAM's CSA limit is exceeded.

> **8**   Expansion would cause the pool to exceed its *xpanlim* specification. Refer to *z/OS Communications Server: SNA Network Implementation Guide* for additional information about *xpanlim*.

> **14**   Not enough CSA storage is available for the expansion.

> **15**   VTAM cannot fix pages in storage due to insufficient page frames or some other page locking problem.

> **17**   Storage unavailable. VTAM's CSA limit is exceeded.

> **18**   Expansion would cause the pool to exceed its *xpanlim* specification. Refer to *z/OS Communications Server: SNA Network Implementation Guide* for additional information about *xpanlim*.

> Refer to the explanation of IST154I in *z/OS Communications Server: SNA Messages* for more information on interpreting byte 16.

17  Flag byte

**Bit**   **Meaning**

**x... ....**
> 1=ISTORFBA caused the expansion.

> 0=ISTORAPX caused the expansion.

**.x.. ....**
> 1=PXB was allocated by this expansion.

> 0=PXB already existed from prior expansion.

..x. ....
    1=Failure to obtain or fix storage.

    0=Success in obtaining or fixing storage.
**18–1B**   Total number of buffers in pool after this expansion (BPCBTOTL)
**1C–1F**   Total number of available buffers in pool after this expansion

---

# EXPP entry for buffer pool expansion

**Entry:  EXPP**
**VIT option:**
        CSM
**Event:**  Pool expansion
**VIT processing module:**
        ISTITCCS
**Control is returned to:**
        IVTSMCEX

This trace record is written when a CSM pool is expanded.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| EXPP | ID   0 | POOL ADDRESS | EXTENT ADDRESS | NUMBER OF BUFFERS IN EXTENT | TOTAL BUFFERS AFTER EXPAND | FREE BUFFERS AFTER EXPAND | 0 |

**Byte (hex)**
        **Contents**
**00–03**   Record ID: C"EXPP"
**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07**   0
**08–0B**   Pool address
**0C–0F**   Extent address
**10–13**   Number of buffers in extent
**14–17**   Total number of buffers in the pool after expansion
**18–1B**   Number of free buffers in the pool after expansion
**1C–1F**   0

---

# FBLK entry for FREEBLK macro (Part 1)

**Entry:  FBLK**
**VIT option:**
        SMS
**Event:**  FREEBLK macro
**VIT processing module:**
        ISTRACSM
**Control is returned to:**
        ISTORCFB or ISTORCDF

This trace record shows the status of each FREEBLK request issued by VTAM components.

The FREEBLK macro is the complement of the GETBLK macro. FREEBLK must release the storage obtained by GETBLK. Each GBLK entry should eventually have a corresponding FBLK entry.

If the return code is nonzero, this entry is generated whether the SMS option is in effect or not. This event is treated as an exception condition and, therefore, is traced whenever the VIT is active.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| FBLK | ID | RTNCD | POOL | VTALC | STORAGE ADDRESS OR 0 | ADDRESS OF SPTAE OR DSPSP | RETURN ADDRESS | LENGTH OF STORAGE FREED | CALLER OF UTILITY OR 0 | RPH ADDRESS |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"FBLK"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code

**06** If FBLK is followed by FBL2:

    **00** AMUTSCBS
    **01** AMUTSCBL
    **02** CMPTAB
    **03** LNKDSTAB
    **04** MNPSRECV
    **05** ICV29CMP

If FBLK is not followed by FBL2, possible storage pool types (in hexadecimal format) follow (refer to *z/OS Communications Server: SNA Network Implementation Guide* for more information on storage pools):

| | | | |
|---|---|---|---|
| 00 RUPEPRIV | 22 UTILCSAL | 44 WAR | 66 BFRTRFUL |
| 01 RUPECOMM | 23 AMU | 45 UVRPL | 67 SLENT |
| 02 SIB | 24 HSICB | 46 DCX | 68 DYPATH |
| 03 SSCPFMCB | 25 LMTABLE | 47 PLUSDATA | 69 PCDCA |
| 04 NQDAT | 26 SAB | 48 ADJCP | 6A XNINFO |
| 05 EPTDVT | 27 RAB | 49 ATGB | 6B GRINS |
| 06 CDRSC | 28 PRIDBLK | 4A TGP | 6C BSBEXT |
| 07 ACDEB | 29 PRIDQAB | 4B KEYTOKEN | 6D SOCKET |
| 08 HSQH | 2A AUTOLOGN | 4C TRSINFO | 6E MIWKE |
| 09 ERTE | 2B CPWACSA | 4D COS | 6F IAP |
| 0A WREEID | 2C PGIOBLK | 4E NDREC | 70 LIA |
| 0B FMCBEXT | 2D PRDLE | 4F TGREC | 71 IPWKE |
| 0C SIBEXT | 2E RIBRANT | 50 ACPCB | 72 VRDCB |
| 0D (Not used) | 2F CANT | 51 DECB | 73 UNSOL |
| 0E UECB | 30 CAB | 52 NIDCB | 74 COWE |
| 0F IOBLOCK | 31 CNSFACUD | 53 CPRUPE | 75 MARB |
| 10 SRTE | 32 BFRTRACE | 54 ANDCB | 76 VRRSB |
| 11 ISTTRCEL | 33 DMTSQ | 55 DISKIO | 77 DDEL |
| 12 UTILPVTS | 34 FMCB | 56 DSERVER | 78 SOCCBEXT |
| 13 VRPL | 35 PLUSFMCB | 57 ADJNODE | 79 RTPINFO |
| 14 POWEPRIV | 36 PXBFIXED | 58 CACHE | 7A CMIPPVT |
| 15 POWECOMM | 37 PXBPAGED | 59 ISTSITCB | 7B PVTSTATC |
| 16 PULURDTE | 38 PLUSC | 5A ISTENDEL | 7C PAGBLBSB |
| 17 PAQ | 39 NSSCB | 5B CORCB | 7D TIPACX |
| 18 RAQ | 3A (Not used) | 5C LCB | 7E CMOBJ |

```
| 19 CPWAPVT    3B (Not used)  5D OSCB      7F CFSPRIV
| 1A ERICPOOL   3C (Not used)  5E SCCB      80 CFSCSA
| 1B SIBIX      3D FMH5        5F DSUTIL    81 SPTPOOL
| 1C CDAJSCP    3E OOBTSCB     60 PLOCB     82 HPRINFO
| 1D GWNAJSCP   3F SLD         61 TREEBLD   84 IPADDR
| 1E IOSIB      40 NSRUS       62 IOBLOCKL  85 IOBLOCKP
| 1F DSSIB      41 NSRUL       63 POWMPRIV
| 20 UTILPVTL   42 RUCON       64 POWMCOMM
| 21 UTILCSAS   43 STB         65 POAPRIV
```

**07**     When set to 1, indicates that storage obtained through GETBLK request and converted to VTALLOC request is returned to system

**08–0B**   Address of block freed (or 0 if FREEBLK failed)

**0C–0F**   Address of storage pool anchor block (SPTAE) or, if FBLK is followed by FBL2, address of DSPSP

**10–13**   Address of the issuer of the FREEBLK macro

**14–17**   Length of storage actually freed not including the 8-byte header

**18–1B**   Caller of utility routine or 0. If the FREEBLK macro was issued from a utility routine, the address of the utility's caller is placed here. A 0 address indicates that the macro was issued directly by the caller (see return address) without a utility routine.

**1C-1F**   Request parameter header (RPH) address

# FBL2 entry for FREEBLK macro (Part 2)

**Entry: FBL2**

**VIT option:**
> SMS

**Event:** FREEBLK macro

**VIT processing module:**
> ISTRACSM

This trace record is a continuation of the FBLK entry; it is generated only if the storage that is freed is in a data space.

FBL2 returns the address of the data space descriptor control block (DSDCB) and the data space name from the DSDCB. When using IPCS, the data space name is required to look at data in the data space.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| FBL2 | DSDCB ADDRESS | DATA SPACE NAME | 0 |

**Byte (hex)**
> **Contents**

**00–03**   Record ID: C"FBL2"

**04—07**   Address of DSDCB

**08—0F**
> Data space name from the DSDCB

**10—1F**
> 0

## FIXB entry for FIX_BUFFER requests

**Entry:** FIXB
**VIT option:**
>   CSM
**Event:** IVTCSM REQUEST=FIX_BUFFER
**VIT processing module:**
>   ISTITCCS
**Control is returned to:**
>   IVTSMCBF

This trace record provides the status of an IVTCSM REQUEST=FIX_BUFFER macroinstruction.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FIXB | | | | ID | 0 | FLAG | RECNUM | 0 | | | | RTN COD | | RSN COD | | UTILRTN CALLER OR RETURN ADDRESS | | | | LAST BUFFER LIST ENTRY | | | | NUMBER OF BUFFERS | | | | THREAD VALUE OR 0 | | | |

**Byte (hex)**

>   **Contents**

**00–03**   Record ID: C"FIXB"

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**   0

**06**   Trace record flag:

>   **B'0... ....'**
>>   Indicates that this is the last record for this event.

>   **B'1... ....'**
>>   Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07**   Trace record number to correlate all the entries for this particular event.

**08–0B**   0

**0C–0D**
>   Return code

**0E–0F**   Reason code

**10–13**   Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

**14–17**   Address of the buffer list entry that was being processed when the error was encountered

**18–1B**   Number of buffers to be fixed

**1C–1F**   THREAD value if specified or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

## FIX2 entry for FIX_BUFFER requests

**Entry:** FIX2

**VIT option:**
　　　CSM
**Event:** IVTCSM REQUEST=FIX_BUFFER
**VIT processing module:**
　　　ISTITCCS

This trace record is a continuation of the FIXB trace record. Each FIX2 record traces, at most, two buffers that were requested to be placed in a fixed state.

| 0 0 0 0 0 0 0 0<br>0 1 2 3 4 5 6 7 | 0 0 0 0 0 0 0 1 1 1 1<br>8 9 A B C D E F 0 1 2 3 | 1 1 1 1 1 1 1 1 1 1 1 1<br>4 5 6 7 8 9 A B C D E F |
|---|---|---|
| FIX2　0　F R<br>L E<br>A C<br>G N<br>U<br>M | BUFFER TOKEN FOR INPUT<br>BUFFER LIST ENTRY | BUFFER TOKEN FOR INPUT<br>BUFFER LIST ENTRY OR 0 |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"FIX2"
**04–05** 0
**06** Trace record flag:

　　**B'0... ....'**
　　　　Indicates that this is the last record for this event.

　　**B'1... ....'**
　　　　Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.
**07** Trace record number to correlate all the entries for this particular event.
**08–13** Buffer token contained in input buffer list entry
**14–1F** Buffer token contained in input buffer list entry or 0

# FRBF entry for FREE_BUFFER requests

**Entry:** **FRBF**
**VIT option:**
　　　CSM
**Event:** IVTCSM REQUEST=FREE_BUFFER
**VIT processing module:**
　　　ISTITCCS
**Control is returned to:**
　　　IVTSMCBF

This trace record provides the status of an IVTCSM REQUEST=FREE_BUFFER macroinstruction.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D | 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FRBF | ID | 0 | FLAG | RECNUM | 0 | RTN COD | RSN COD | UTILRTN CALLER OR RETURN ADDRESS | LAST BUFFER LIST ENTRY | NUMBER OF BUFFERS | THREAD VALUE OR 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"FRBF"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace record flags:

> **B'0... ....'**
> Indicates that this is the last record for this event.

> **B'1... ....'**
> Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07** Trace record number to correlate all the entries for this particular event.

**08–0B** 0

**0C–0D**
> Return code

**0E–0F** Reason code

**10–13** Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

**14–17** Address of the buffer list entry that was being processed when the error was encountered

**18–1B** Number of buffers requested to be freed

**1C–1F** THREAD value if specified or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

## FRB2 entry for FREE_BUFFER requests

**Entry: FRB2**
**VIT option:**
> CSM

**Event: IVTCSM REQUEST=FREE_BUFFER**
**VIT processing module:**
> ISTITCCS

This trace record is a continuation of the FRBF trace record. It contains information about the buffers that were freed. Each FRB2 record traces, at most, two buffers that were freed.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| FRB2 | | 0 | F L A G / R E C N U M | BUFFER TOKEN FOR INPUT BUFFER LIST ENTRY | | | BUFFER TOKEN FOR INPUT BUFFER LIST ENTRY OR 0 | | |

**Byte (hex)**

**Contents**

**00–03**   Record ID: C"FRB2"

**04–05**   0

**06**   Trace record flag:

**B'0... ....'**
> Indicates that this is the last record for this event.

**B'1... ....'**
> Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07**   Trace record number to correlate all the entries for this particular event

**08–13**   Buffer token contained in input buffer list entry

**14–1F**   Buffer token contained in input buffer list entry or 0

# FRES entry for FREESTOR macro

**Entry:  FRES**
**VIT option:**
> SMS

**Event:**  Invocation of the FREESTOR macro
**VIT processing module:**
> ISTRACSM

**Control is returned to:**
> Module invoking the FREESTOR macro

This trace record shows the status of each FREESTOR request issued by VTAM components. The FREESTOR macro releases the storage obtained by the GETSTOR macro. Each GETS entry should eventually have a corresponding FRES entry.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F |
| FRES | I D | 0 | R T N C O D E / I N D E X | STORAGE ADDRESS | ADDRESS OF ORGSH | RETURN ADDRESS | A L L O C L E N | 0 | RETURN ADDRESS OF CALLER | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"FRES"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Return code

**07** Pool Index (in hex)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 00 | CFSACCPS | 07 | NLPDELPD | 0E | CSAFXVNE | 15 | PVTPGVNT |
| 01 | CFSACCPD | 08 | TCPIOCD | 0F | CSAFXJFE | 16 | CFSBUFPS |
| 02 | CFSACCCS | 09 | CSAPGVFE | 10 | CSAFXJNE | 17 | CFSBUFPD |
| 03 | CFSACCCD | 0A | CSAPGVNE | 11 | PVTPGJFJ | 18 | CFSBUFCS |
| 04 | RPMNPSPS | 0B | CSAPGJFE | 12 | PVTPGJFT | 19 | CFSBUFCD |
| 05 | HIPOOLPS | 0C | CSAPGJNE | 13 | PVTPGJNT | 1A | ALPHCD |
| 06 | MRPOOLPS | 0D | CSAFXVFE | 14 | PVTPGVFT | 1B | EEHNMIPD |

**08–0B** Address of storage freed

**0C–0F** Address of GETSTOR header (ISTORGSH)

**10–13** Address of the invoker of FREESTOR macro

**14–15** Size of pool

**16–17** 0

**18–1B** Caller of utility routine or 0. If the FREESTOR macro was issued from a utility routine, the address of the utility's caller is placed here. A 0 address indicates that the macro was issued directly by the caller (see return address) without a utility routine.

**1C–1F** Request parameter header (RPH) address

# GBLK entry for GETBLK macro (Part 1)

**Entry: GBLK**

**VIT option:**
    SMS

**Event:** GETBLK macro

**VIT processing module:**
    ISTRACSM

**Control is returned to:**
    ISTORCDG or ISTORCGB

This trace record shows the status of each GETBLK request issued by VTAM components.

The FREEBLK macro is the complement of the GETBLK macro. FREEBLK must release the storage obtained by GETBLK. Each GBLK entry should eventually have a corresponding FBLK entry.

If the return code is nonzero, this entry is generated whether the SMS option is in effect or not. It is treated as an exception condition and is always traced if the VIT is active, regardless of the SMS option.

| 0 0 0 0 | 0 4 | 0 5 | 0 6 | 0 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| GBLK | I D | R T N C D | P O O L | F L A G S | STORAGE ADDRESS OR 0 | ADDRESS OF SPTAE OR DSPSP | RETURN ADDRESS | LENGTH OF STORAGE REQUESTED \| OBTAINED OR ZERO | CALLER OF UTILITY OR 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"GBLK"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code

**06** If GBLK is followed by GBL2:

    **00**    AMUTSCBS

    **01**    AMUTSCBL

    **02**    CMPTAB

    **03**    LNKDSTAB

    **04**    MNPSRECV

    **05**    ICV29CMP

If GBLK is not followed by GBL2, possible storage pool types (in hexadecimal format) follow (refer to *z/OS Communications Server: SNA Network Implementation Guide* for more information on storage pools):

| | | | |
|---|---|---|---|
| 00 RUPEPRIV | 22 UTILCSAL | 44 WAR | 66 BFRTRFUL |
| 01 RUPECOMM | 23 AMU | 45 UVRPL | 67 SLENT |
| 02 SIB | 24 HSICB | 46 DCX | 68 DYPATH |
| 03 SSCPFMCB | 25 LMTABLE | 47 PLUSDATA | 69 PCDCA |
| 04 NQDAT | 26 SAB | 48 ADJCP | 6A XNINFO |
| 05 EPTDVT | 27 RAB | 49 ATGB | 6B GRINS |
| 06 CDRSC | 28 PRIDBLK | 4A TGP | 6C BSBEXT |
| 07 ACDEB | 29 PRIDQAB | 4B KEYTOKEN | 6D SOCKET |
| 08 HSQH | 2A AUTOLOGN | 4C TRSINFO | 6E MIWKE |
| 09 ERTE | 2B CPWACSA | 4D COS | 6F IAP |
| 0A WREEID | 2C PGIOBLK | 4E NDREC | 70 LIA |
| 0B FMCBEXT | 2D PRDLE | 4F TGREC | 71 IPWKE |
| 0C SIBEXT | 2E RIBRANT | 50 ACPCB | 72 VRDCB |
| 0D (Not used) | 2F CANT | 51 DECB | 73 UNSOL |
| 0E UECB | 30 CAB | 52 NIDCB | 74 COWE |
| 0F IOBLOCK | 31 CNSFACUD | 53 CPRUPE | 75 MARB |
| 10 SRTE | 32 BFRTRACE | 54 ANDCB | 76 VRRSB |
| 11 ISTTRCEL | 33 DMTSQ | 55 DISKIO | 77 DDEL |
| 12 UTILPVTS | 34 FMCB | 56 DSERVER | 78 SOCCBEXT |
| 13 VRPL | 35 PLUSFMCB | 57 ADJNODE | 79 RTPINFO |
| 14 POWEPRIV | 36 PXBFIXED | 58 CACHE | 7A CMIPPVT |
| 15 POWECOMM | 37 PXBPAGED | 59 ISTSITCB | 7B PVTSTATC |
| 16 PULURDTE | 38 PLUSC | 5A ISTENDEL | 7C PAGBLBSB |
| 17 PAQ | 39 NSSCB | 5B CORCB | 7D TIPACX |
| 18 RAQ | 3A (Not used) | 5C LCB | 7E CMOBJ |
| 19 CPWAPVT | 3B (Not used) | 5D OSCB | 7F CFSPRIV |
| 1A ERICPOOL | 3C (Not used) | 5E SCCB | 80 CFSCSA |

| | | | |
|---|---|---|---|
| 1B SIBIX | 3D FMH5 | 5F DSUTIL | 81 SPTPOOL |
| 1C CDAJSCP | 3E OOBTSCB | 60 PLOCB | 82 HPRINFO |
| 1D GWNAJSCP | 3F SLD | 61 TREEBLD | 84 IPADDR |
| 1E IOSIB | 40 NSRUS | 62 IOBLOCKL | 85 IOBLOCKP |
| 1F DSSIB | 41 NSRUL | 63 POWMPRIV | |
| 20 UTILPVTL | 42 RUCON | 64 POWMCOMM | |
| 21 UTILCSAS | 43 STB | 65 POAPRIV | |

**07**  Flags:

| Bit | Meaning |
|---|---|
| **1... ....** | Length of storage requested and length of storage obtained are in GBL2 continuation |
| **.1.. ....** | Length of storage requested and length of storage obtained are in GBL3 continuation |
| **..xx xxx.** | 0 |
| **.... ...1** | GETBLK request converted to VTALLOC request |

**08–0B**  Address of block obtained (or 0 if GETBLK failed)

**0C–0F**  Address of storage pool anchor block (SPTAE) or, if GBLK is followed by GBL2, address of DSPSP

**10–13**  Address of the issuer of the GETBLK macro

**14–15**  If byte 07, bits 0 and 1 are 0, length of storage specified by the user. Otherwise, 0.

**16-17**  If byte 07, bits 0 and 1, are 0, length of storage actually obtained, rounded to the correct subpool length. Otherwise, 0. This length does not include the 8-byte GETBLK header.

**18-1B**  Caller of utility routine or 0. If the GETBLK macro was issued from a utility routine, the address of the utility's caller is placed here. A 0 address indicates that the macro was issued directly by the caller (see return address) without a utility routine.

**1C-1F**  Request parameter header (RPH) address

## GBL2 entry for GETBLK macro (Part 2)

**Entry:  GBL2**
**VIT option:**
SMS
**Event:**  GETBLK macro
**VIT processing module:**
ISTRACSM

This trace record is a continuation of the GETBLK entry and provides the address of the data space descriptor control block (DSDCB). This trace record is generated only if the storage that is acquired is in a data space.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F |
| GBL2 | DSDCB ADDRESS | DATA SPACE NAME | LENGTH OF STORAGE REQUESTED OR ZERO | LENGTH OF STORAGE OBTAINED OR ZERO | 0 |

**Byte (hex)**
**Contents**

00–03    Record ID: C"GBL2"
04–07    Address of DSDCB
08–0F    Data space name from the DSDCB. When you are using IPCS, the data space name is required often to look at data in the data space.
10–13    If byte 07, bit 0 of GBLK record is 1, length of storage specified by the user. Otherwise, 0.
14-17    If byte 07, bit 0 of GBLK record is 1, length of storage actually obtained, rounded to the correct subpool length. Otherwise, 0. This length does not include the 8-byte GETBLK header.
18-1F    0

## GBL3 entry for GETBLK macro (Part 2)

**Entry:  GBL3**
**VIT option:**
>    SMS
**Event:  GETBLK macro**
**VIT processing module:**
>    ISTRACSM

This trace record is a continuation of the GETBLK entry and provides the length of storage requested and the length of storage obtained if either length is 65 536 or greater. This trace record is generated when needed to provide lengths of that magnitude, but only if the storage that is acquired is not in a data space.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GBL3 | | | | 0 | | | | | | | | | | | | LENGTH OF STORAGE REQUESTED | | | | LENGTH OF STORAGE OBTAINED | | | | 0 | | | | | | | |

**Byte (hex)**
>    **Contents**
00–03    Record ID: C"GBL3"
04–0F    0
10-13    Length of the storage specified by the user.
14-17    Length of storage actually obtained, rounded to the correct subpool length. This length does not include the 8-byte GETBLK header.
18-1F    0

## GCEL entry for Get Cell requests

**Entry:  GCEL**
**VIT option:**
>    CIA
**Event:  CSM storage acquisition**
**VIT processing module:**
>    ISTRACCI
**Control is returned to:**
>    IUTLLCCC (Module that issued the request)

This trace record shows the status of each (Get Cell) request for a CSM cell. It captures the outcome of the CELLCTL FUNC(GET) macro. This macro is used by DLC components.

There is no corresponding free cell event. The cells are freed directly to CSM using IVTCSM REQUEST=FREE_BUFFER (see "FRBF entry for FREE_BUFFER requests" on page 195) and are normally freed by non-VTAM components.

| 0 0 0 0 | 0 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| GCEL | ID 0 | RET CD OR LEN | CSM BUFFER ADDR | ADDRESS OF CALLER | CSM BUFFER TOKEN | RPH |

**Byte (hex)**
 **Contents**
**01–03** Record ID: C"GCEL"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06–07** Return code or length
**08–0B** CSM buffer address
**0C–0F** Address of caller
**10–1B** CSM buffer token
**1C–1F** RPH address

# GCEx entry for CSM storage movement

**Entry:** GCEA or GCER
**VIT option:**
 CIA
**Event:** CSM storage movement
**VIT processing module:**
 ISTRECCI
**Control is returned to:**
 The module that issued the INTRACE

This trace record shows the movement of CSM cells between different queues. This macro is used for QDIO and HiperSockets™ devices.

| 0 0 0 0 | 0 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 | A B | C D E F |
| GCEA GCER | ID 0 | CNT | ADDR OF CELL | ADDR OF CALLER | ADDR OF QUEUE ANCHOR | ANCHOR HEAD | MOD EYE | 0 | RPH OR THREAD |

**Byte (hex)**
 **Contents**
**00–03** Record ID:
  C"GCEA" for adding CSM cell to queue
  C"GCER" for removing CSM cell from queue
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**      0
**06–07**  Count of buffers available on queue
**08–0B**  CSM buffer address
**0C–0F**  Address of caller
**10–13**  Starting address of the queue that the CSM cell is being added to or removed from
**14–17**  First four bytes of anchor
**18–19**  Module eye catcher of caller
**1A–1B**  0
**1C–1F**  RPH address or thread

# GETS entry for GETSTOR macro

**Entry: GETS**
**VIT option:**
        SMS
**Event:** Invocation of the GETSTOR macro
**VIT processing module:**
        ISTRACSM
**Control is returned to:**
        Module invoking the GETSTOR macro

This trace record shows the status of each GETSTOR request issued by VTAM components.

A FREESTOR macro releases the storage obtained by GETSTOR. Each GETS entry should eventually have a corresponding FRES entry.

| 0 0 0 0<br>0 1 2 3 | 0<br>4 | 0<br>5 | 0<br>6 | 0<br>7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1 1 1<br>0 1 2 3 | 1 1 1 1<br>4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| GETS | I<br>D | 0 | R<br>T<br>N<br><br>C<br>O<br>D<br>E | I<br>N<br>D<br>E<br>X | STORAGE<br>ADDRESS | ADDRESS<br>OF<br>ORGSH | RETURN<br>ADDRESS | A<br>L<br>L<br>O<br>C<br><br>L<br>E<br>N | RETURN<br>ADDRESS<br>OF<br>CALLER | RPH<br>ADDRESS |

**Byte (hex)**
        **Contents**
**00–03**  Record ID: C"GETS"
**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**      0
**06**      Return code
**07**      Pool Index (in hex)

| 00 | CFSACCPS | 07 | NLPDELPD | 0E | CSAFXVNE | 15 | PVTPGVNT |
|---|---|---|---|---|---|---|---|
| 01 | CFSACCPD | 08 | TCPIOCD | 0F | CSAFXJFE | 16 | CFSBUFPS |
| 02 | CFSACCCS | 09 | CSAPGVFE | 10 | CSAFXJNE | 17 | CFSBUFPD |
| 03 | CFSACCCD | 0A | CSAPGVNE | 11 | PVTPGJFJ | 18 | CFSBUFCS |
| 04 | RPMNPSPS | 0B | CSAPGJFE | 12 | PVTPGJFT | 19 | CFSBUFCD |
| 05 | HIPOOLPS | 0C | CSAPGJNE | 13 | PVTPGJNT | 1A | ALPHCD |
| 06 | MRPOOLPS | 0D | CSAFXVFE | 14 | PVTPGVFT | 1B | EEHNMIPD |

**08–0B**  Address of storage returned

0C–0F   Address of GETSTOR header (ISTORGSH)
10–13   Address of the invoker of GETSTOR macro
14–17   Allocated storage length
18–1B   Caller of utility routine. If the GETSTOR macro was issued from a utility
routine, the address of the utility's caller is placed here. A 0 address
indicates that the macro was issued directly by the caller (see return
address) without a utility routine.
1C–1F   Request parameter header (RPH) address

# GNAM entry for GNAME macro invoked (Part 1)

**Entry:  GNAM**
**VIT option:**
SSCP
**Event:**  GNAME macro invocation
**VIT processing module:**
ISTRACSC
**Control is returned to:**
ISTCPCGM

This trace record is written when generic mapping information in the generic
resource coupling facility structure is created, deleted, or changed using the
GNAME macroinstruction.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F | 10 11 12 13 | 14 | 15 16 | 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| GNAM | ID | RTNCD | FUNC | FLAGS | GENERIC RESOURCE NAME | RETURN ADDRESS | CODE | 0 | RCNT | GENERIC NUMBER | RPH ADDRESS |

**Byte (hex)**
**Contents**
00–03   Record ID: C"GNAM"
04   ID is the primary address space ID (ASID). This field is 0 if the ASID is
greater than X'FF'.
05   Return code from the GNAME macro
**X'00'**   Request completed successfully. This is an existing USERVAR
return code.
**X'01'**   Request completed successfully and local data was returned. This
is only returned from GNAME if SEARCH is CF&LOCAL.
**X'04'**   Resource specified could not be located. This is an existing
USERVAR return code.
**X'51'**   Request failed due to APPC restrictions. This is an existing
USERVAR return code.
**X'52'**   Request failed due to a parameter specification that was not valid.
This is an existing USERVAR return code.
**X'53'**   The function requested could not be completed because of a
storage allocation failure. This is an existing USERVAR return code.
**X'54'**   The real instance of user variable to be deleted could not be
located. This is an existing USERVAR return code.
**X'55'**   A conflict was found due to NETIDs.
**X'56'**   A conflict was found due to CPNAMEs.

| | | |
|---|---|---|
| | **X'57'** | VTAM is halting. |
| | **X'58'** | ENDAFFINITY was issued for a session that was VTAM owned. |
| | **X'59'** | RSCTYPE value conflicted between USERVAR and GR. |
| | **X'5A'** | An attempt to change the generic name for a given application failed because SPTEs pertaining to the previous generic name still exist. |
| | **X'60'** | A session pair could not be found in a SPT entry. |
| | **X'61'** | An attempt to change the generic name for a given application failed because either (1) the existing generic name was never deleted, or (2) because SPTEs pertaining to the existing generic name still exist. |
| | **X'62'** | An attempt to repopulate the generic resource coupling facility structure has failed because the local data being used for repopulation is back level compared to data already in the structure. |
| | **X'63'** | An attempt to repopulate the generic resource coupling facility structure has succeeded but backlevel data has been identified in the structure and additional structure cleanup may be required. |
| | **X'64'** | An attempt to increment the session count for a SPTE failed because the name type of the SPTE did not match what was expected. |
| | **X'65'** | Request failed due to TSO mismatched. |
| | **X'66'** | An attempt to register a generic resource with the work load manager failed due to the use of an STOKEN that is not valid. |
| | **X'A0'** | A request made against the generic resources coupling facility structure will complete synchronously. |
| | **X'A1'** | An attempt to update information in the generic resources coupling facility structure failed because that data had changed since it was last read. The data should be re-read and then modified again. |
| | **X'A2'** | An attempt to access the generic resources coupling facility structure failed for an unexpected reason. |
| | **X'A3'** | There is currently no connection to the generic resources coupling facility structure. |
| | **X'A4'** | The buffers provided for reading data from the generic resources coupling facility structure were insufficient for buffering all the data associated with the list entry being read. No data (adjunct or element) is returned. |
| | **X'A5'** | A read from the generic resources coupling facility structure failed because the requested data could not be found in that structure. |
| | **X'A6'** | Data could not be added to the generic resources coupling facility structure because there is insufficient storage in the generic resources coupling facility structure to hold it. |
| **06** | | Reason macro invoked: |
| | **X'01'** | Find generic resource mapping. |
| | **X'02'** | Find generic resource mapping or USERVAR. |
| | **X'03'** | Add the application program network name to generic mapping. |
| | **X'04'** | Delete the application program network name from generic mapping. |
| | **X'05'** | Write generic mapping to coupling facility structure. |
| | **X'06'** | Free local copy of generic mapping. |
| | **X'07'** | Find a generic resource name for an application program network name. |
| | **X'08'** | Find the generic number for a generic resource name. |
| **07** | | Event flags: |

| Bit | Meaning |
|---|---|
| **1... ....** | The addition or deletion of an application program network name is due to a change in CP-CP status. |
| **.1.. ....** | The addition or deletion of an application program network name is due to a SETLOGON GNAMEADD or SETLOGON GNAMEDEL. |
| **..1. ....** | Partner is local, a local real instance will be given precedence (used only for resolution. See byte 14). |
| **...1 ....** | Resolve to this node, the real instance must be on this node (used only for resolution - see byte 14). |
| **.... 1...** | Update resolution count, the resolution count will be incremented or decremented (used only for resolution and termination - see byte 14). |
| **.... .1..** | Update session count, the session count will be incremented or decremented (used only for resolution and termination - see byte 14). |
| **.... ..1.** | Real instance is a subordinate resource. |
| **.... ...x** | 0 |

| Byte | Contents |
|---|---|
| **08–0F** | Generic name |
| **10–13** | Address of the invoker of the GNAME macro |
| **14** | Reason code (used for reason macros X'01' and X'05') |

| | |
|---|---|
| **X'00'** | Resolution |
| **X'01'** | Reserved |
| **X'02'** | Update session counts |
| **X'03'** | Termination |

| Byte | Contents |
|---|---|
| **15–16** | 0 |
| **17** | Resolution count or 0 |
| **18–1B** | The generic number if available, otherwise 0 |
| **1C–1F** | Request parameter header (RPH) address |

## GNA2 entry for GNAME macro invoked (Part 2)

**Entry: GNA2**
**VIT option:**
    SSCP
**Event:** GNAME macro invocation
**VIT processing module:**
    ISTRACSC

This trace record is a continuation of the GNAM entry.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| GNA2 | SESS COUNT OR 0 | PCID | NETID OF APPLICATION PROGRAM NETWORK NAME | APPLICATION PROGRAM NETWORK NAME |

**Byte (hex)**
    **Contents**

| Byte | Contents |
|---|---|
| **00–03** | Record ID: C'GNA2' |
| **04–07** | Session count or 0 |
| **08–0F** | PCID if associated with a session, otherwise 0 |

10–17    Network ID of application program network name
18–1F    Application program network name

___

# GTBF entry for GET_BUFFER requests

**Entry: GTBF**
**VIT option:**
    CSM
**Event:** IVTCSM REQUEST=GET_BUFFER
**VIT processing module:**
    ISTITCCS
**Control is returned to:**
    IVTSMCBF

This trace record provides the status of IVTCSM REQUEST=GET_BUFFER macroinstructions.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 | 09 0A 0B | 0C 0D | 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GTBF | ID | 0 | FLAG | RECNUM | POOL | 0 | RTN COD | RSN COD | UTILRTN CALLER OR RETURN ADDRESS | LAST BUFFER LIST ENTRY | 0 | THREAD VALUE OR 0 |

**Byte (hex)**
    **Contents**
**00–03**    Record ID: C"GTBF"
**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**    0
**06**    Trace record flag:

    **B'0... ....'**
        Indicates that this is the last record for this event.

    **B'1... ....'**
        Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.
**07**    Trace record number to correlate all the entries for this particular event
**08**    Pool identifier, can be one of the following values:
    **X'10'**        4K ECSA pool
    **X'20'**        16K ECSA pool
    **X'30'**        32K ECSA pool
    **X'40'**        60K ECSA pool
    **X'60'**        180K ECSA pool
    **X'90'**        4K 31-bit backed DSPACE pool
    **X'98'**        4K 64-bit backed DSPACE pool
    **X'A0'**        16K 31-bit backed DSPACE pool
    **X'A8'**        16K 64-bit backed DSPACE pool
    **X'B0'**        32K 31-bit backed DSPACE pool
    **X'B8'**        32K 64-bit backed DSPACE pool
    **X'C0'**        60K 31-bit backed DSPACE pool

| | | |
|---|---|---|
| **X'C8'** | 60K 64-bit backed DSPACE pool | |
| **X'E0'** | 180K 31-bit backed DSPACE pool | |
| **X'E8'** | 180K 64-bit backed DSPACE pool | |

**09–0B**   0

**0C–0D**
        Return code

**0E–0F**   Reason code

**10–13**   Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

**14–17**   Address of the buffer list entry that was being processed when the error was encountered.

**18–1B**   0

**1C–1F**   THREAD value if specified or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

# GTB2 entry for GET_BUFFER requests

**Entry: GTB2**

**VIT option:**
        CSM

**Event:** IVTCSM REQUEST=GET_BUFFER

**VIT processing module:**
        ISTITCCS

This trace record is a continuation of the GTBF trace record. It contains additional information about the IVTCSM REQUEST=GET_BUFFER macroinstruction.

| 00 01 02 03 | 04 05 06 | 07 | 08 | 09 | 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 | 16 17 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| GTB2 | 0 | F L A G | R E C N U M | T Y P E | S O U R C E | O W N E R | I D | NUMBER OF BUFFERS | FREE ROUTINE ADDRESS |
| | | | | | | | | 0 | POOL TOKEN |

**Byte (hex)**

    **Contents**

**00–03**   Record ID: C"GTB2"

**04–05**   0

**06**       Trace record flag:

        **B'0... ....'**
                Indicates that this is the last record for this event.

        **B'1... ....'**
                Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07**       Trace record number to correlate all the entries for this particular event

**08**       Buffer type

        **X'20'**    Page-eligible

        **X'40'**    Pageable

| | | |
|---|---|---|
| **X'80'** | Fixed | |
| **09** | Buffer source | |
| | **X'40'** | DSPACE pool |
| | **X'80'** | ECSA pool |
| **0A–0B** | OWNERID | |
| **0C–0F** | Number of buffers requested | |
| **10–13** | Free routine address | |
| **14–15** | 0 | |
| **16–1F** | Pool token | |

## GTB3 entry for GET_BUFFER requests

**Entry: GTB3**
**VIT option:**
    CSM
**Event:** IVTCSM REQUEST=GET_BUFFER
**VIT processing module:**
    ISTITCCS

This record is written for a CSM buffer allocated as a result of the IVTCSM REQUEST=GET_BUFFER macroinstruction. This trace record corresponds to an entry in the buffer list returned by the IVTCSM macroinstruction.

| 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 4 5 6 7 | 8 9 A B C D E F 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| GTB3   0   FLAG   RECNUM | BUFFER TOKEN | BUFFER ALET | BUFFER ADDRESS | BUFFER SIZE |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"GTB3"
**04–05** 0
**06** Trace record flag:

    **B'0... ....'**
        Indicates that this is the last record for this event.

    **B'1... ....'**
        Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.
**07** Trace record number to correlate all the entries for this particular event
**08–13** Buffer token
**14–17** Buffer ALET
**18–1B** Buffer address
**1C–1F** Buffer size

## HCLK entry for HPR clock event

**Entry: HCLK**

**VIT option:**
> HPR

**Event:** HPR clock event

**VIT processing module:**
> ISTITCHP

**Control is returned to:**
> ISTRPCTM

This trace record is written when the clock state changes (approximately every second). This trace record is also written when the clock mode changes.

| 0 0 | 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| HCLK | | A S I D | 0 | O L D S T A T E | N E W S T A T E | TIME STAMP | NUMBER OF 25 ms TIMERS ON CLOCK | NUMBER OF LIVENESS TIMERS | NUMBER OF TIMERS ON LATE QUEUE | RPH ADDRESS |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"HCLK"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Old HPR clock state
> **P**      Pending stop
> **R**      Running
> **S**      Stopped

**07** New HPR clock state
> **P**      Pending stop
> **R**      Running
> **S**      Stopped

**08–0F** Time stamp of HCLK entry

**10–13** Total number of 25 millisecond timers currently on the HPR clock

**14–17** Number of liveness timers on the HPR clock

**18–1B** Number of timers on the late queue

**1C–1F** Request parameter header (RPH) address

## HCL2 entry for HPR clock event (Part 2)

**Entry:** HCL2

**VIT option:**
> HPR

**Event:** HPR clock event

**VIT processing module:**
> ISTITCHP

**Control is returned to:**
> ISTRPCTM

This trace record is a continuation of the HCLK entry.

| 0 0 | 0 | 0 | 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 0 0 0 <br> 8 9 A B | 0 0 0 0 <br> C D E F | 1 1 1 1 1 1 1 1 <br> 0 1 2 3 4 5 6 7 | 1 1 1 1 <br> 8 9 A B | 1 1 1 1 <br> C D E F |
| HCL2 | | | 0 | | C U R R R A T E | N E X T R A T E | NUMBER OF 1ms TIMERS ON CLOCK | NUMBER OF LIVENESS2 TIMERS FOR ONEHOP EE | 0 | NUMBER OF HPRPST PATH SWITCH TIMERS | NUMBER OF NEW ROUTE PATH SWITCH TIMERS |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C'HCL2'

**04-05**  0

**06**  Current HPR clock rate

    **H**      High mode

    **S**      Standard mode

**07**  Next HPR clock rate

    **H**      High mode

    **S**      Standard mode

**08-0B**  Total number of 1 millisecond timers currently on the HPR clock

**0C-0F**  Number of one-hop EE liveness timers (Liveness2) on the HPR clock

**10-17**  0

**18-1B**  Number of HPRPST path switch timers on the HPR clock

**1C-1F**  Number of new route path switch timers on the HPR clock

# HIOx entry for Halt I/O

**Entry:  HIOB, HIOD, HIOH, HIOI, HIOT, or HIOX**

**VIT option:**

    CIO

**Event:  Halt I/O**

**VIT processing module:**

    ISTRACCI

**Control is returned to:**

    ISTTSCLR

This trace record is written when a Halt I/O SVC is issued to end a currently executing channel program.

- HIOB is generated when an HIO SVC ends a channel program using a BSC line attached to the communication adapter.

- HIOD is generated when an HIO SVC ends a channel program for an OSA-Express QDIO or HiperSockets connection.

- HIOH is generated when an HIO SVC ends a channel program using an SDLC line attached to the communication adapter.

- HIOI is generated when a VM GENIO HALT or an MVS HIO SVC ends a channel program to a local SNA controller as a result of VARY INACT,FORCE processing.

- HIOT is generated when a Halt I/O SVC ends a channel program for a TCP/IP DLC connection.

- HIOX is generated when an HIO SVC ends communication between channel-to-channel-attached hosts.

An HIO entry should be followed shortly by an interrupt (INT) entry. Match the CUAs in the HIO and INT entries to be sure they are for the same device. Refer to *z/OS Communications Server: SNA Data Areas Volume 1* for a description of the NCB, ICNCB, and XCNCB fields.

For record types with suffix I, X, or T, the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| HIOB HIOH HIOI HIOT HIOX | | | | ID | STATE | MODID | | CUA DEVICE | | | | NCB ADDRESS | | | | FLAG BYTES | | | | 0 | | | | | | | | | | DLC OR 0 | 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID:
- C"HIOB"
- C"HIOD"
- C"HIOH"
- C"HIOI"
- C"HIOT"
- C"HIOX"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** For HIOB, link station state (BSCLKFSM)

For HIOD, adapter state (DINCB_AFSM)

For HIOH, link station state (HALLFSM)

For HIOI, link state (NCBLNKST)

For HIOX, for a channel-to-channel adapter, station state (XCNSSFSM)

For HIOX, for a LAN channel station, link state (NCBLNKST)

For HIOT, link state (NCBLNKST)

**06–07** Module ID (last 2 characters of TSC module name)

**08–0B** Channel device name in EBCDIC (either a device address or device number)

**0C–0F** NCB address

**10–13** Flag bytes (NCBFLAGS)

**14–1D** 0

**1E** For HIOB, HIOD, HIOH, HIOI, and HIOX: 0

For HIOT:
- For the CDLC DLC: C
- For the LCS DLC: L
- For the CLAW DLC: W
- For the CTC DLC: X
- For the HYPERchannel DLC: H

**1F** 0

## HLST entry for subtrace tree

**Entry:  HLST**
**VIT option:**
> SSCP

**Subtrace type:**
> TREE

**Event:**  APPN route computation
**VIT processing module:**
> ISTITCAB

**Control is returned to:**
> ISTTRQWC

This record is generated when:

- Subtrace TREE under SSCP trace option is active.
- A candidate hop is processed by TRS in routing tree build or update processing.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HLST | | | | ASID | 0 | INSTANCE | TG NUMBER | TG WEIGHT | NODE WGHT | PATH WGHT | | CURRENT CP NAME | | | | | | | | PARENT CP NAME | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**
> **Contents**

**00–03**   Record ID: C"HLST"
**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**   0
**06**   Trace entry instance
**07**   TG number for the current hop
**08**   TG weight
**09**   Node weight of the current hop
**0A–0B**   Total path weight from the tree root to the current node
**0C–13**   CP name of the destination node on current hop
**14–1B**   CP name of the origin node on current hop
**1C–1F**   Request parameter header (RPH) address

## HLS2 entry for subtrace tree (Part 2)

**Entry:  HLS2**
**VIT option:**
> SSCP

**Subtrace type:**
> TREE

**Event:**  APPN route computation
**VIT processing module:**
> ISTITCAB

**Control is returned to:**
 ISTTRQWC

This record is a continuation of the HLST trace record. HLS2 is generated only if any of the CP names in the HLST record has a different network ID than the local node.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| HLS2 | | | | 0 | | | | | | | | CURRENT NODE NETWORK ID | | | | | | | | PARENT NODE NETWORK ID | | | | | | | | 0 | | | |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"HLS2"
**04–0B** 0
**0C–13** Network ID of the destination node on current hop
**14–1B** Network ID of the origin node on current hop
**1C–1F** 0

# HPR entry for HPRCTL macroinstruction (Part 1)

**Entry: HPR**
**VIT option:**
 HPR
**Event:** Invocation of HPRCTL macroinstruction
**VIT processing module:**
 ISTITCHP
**Control is returned to:**
 Module invoking the HPRCTL macroinstruction

This trace record is written when the HPRCTL macroinstruction is issued. It indicates the type of lookup being performed, the return code for that operation, and the addresses of the parameters involved.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| HPR | | | | I D | 0 | T Y P E | F U N C T I O N | R C | ISSUING MODULE NAME | | | | | I N S T A N C E | 0 | ADDR OR SESSION COUNT | | | | COS NAME OR ADDR OR INDEX | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"HPR"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  0

**06**  Entry type:
   **F**    FID5
   **L**    LU
   **M**    MRSU
   **P**    PMI
   **R**    RTP
   **S**    RSREC
   **T**    RTREC

**07**  Function:
   **A**    Add
   **C**    Change
   **D**    Delete
   **F**    Find
   **P**    Purge
   **S**    Associate

**08**  Return code:
   **00**    Success
   **04**    Not found
   **08**    Insufficient storage
   **0C**    Length value not valid
   **10**    Function not supported
   **14**    Duplicate FID5
   **18**    Token not valid

**09–0D**  Name of module issuing HPRCTL

**0E**  HPRCTL instance within issuing module

**0F**  0

**10–13**
- If entry type is FID5, BSB address
- If entry type is LU, session count
- If entry type is PMI and function is Add or Find, CPNRB address
- If entry type is RSREC, RSREC address
- If entry type is RTP, RTP address
- If entry type is RTREC, RTREC address

**14–1B**
- If entry type is FID5, FID5 address
- If entry type is LU, not applicable

- If entry type is MRSU, the MRSU address is located in bytes X'14' and X'15'
- If entry type is PMI, not applicable
- If entry type is RSREC, element address and element index
- If entry type is RTP, APPN CoS name. It is not written, however, for some cases of FIND FIRST and FIND NEXT.
- If entry type is RTREC, element address and element index

1C–1F   Request parameter header (RPH) address

## HPR2 entry for HPRCTL macroinstruction (Part 2)

**Entry:  HPR2**
**VIT option:**
   HPR
**Event:**  Invocation of HPRCTL macroinstruction
**VIT processing module:**
   ISTITCHP

This trace record is a continuation of the HPR entry.

| 0 0 0 0 | 0 | 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 | 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| HPR2 | 0 | F L A G S | CP NAME OR LU NAME OR RPNCB ADDRESS OR LOCAL NCE LABEL | REMOTE NCE LABEL OR PCID |

**Byte (hex)**
   **Contents**
00–03   Record ID: C"HPR2"
04      0
05–06   HPRCTL flags:

   Byte 1

   **Bit     Meaning**
   **xx.. ....**
           BSB lock level requested (entry type FID5):
           **00**      No lock
           **01**      Shared
           **10**      Exclusive
   **..1. ....**
           Decrement LU session count (entry type LU)

           FIND FIRST (entry type RTP or PMI)
   **...1 ....**
           Increment LU session count (entry type LU)

           FIND NEXT (entry type RTP or PMI)
   **.... 1...**
           LU entry was deleted (entry type LU)
   **.... .1..**
           An unused LU entry was found and deleted (entry type LU)
   **.... ..1.**
           Expired entry request (entry type RSREC)

.... ...1
   APPN boundary function when RTP ends

Byte 2

| Bit | Meaning |
|---|---|
| xxx. .... | HPRCTL entry type |
| 000 | FID5 |
| 001 | LU |
| 010 | RTP |
| 011 | RSREC |
| 100 | MRSU |
| 101 | PMI |
| 110 | RTREC |
| ...x xxxx | Available |

**07–17**

- If entry type is PMI, RSREC, or RTREC, CP name
- If entry type is RTP and HPRCTL flag is FIND FIRST or FIND NEXT, CP name

| Byte | Contents |
|---|---|
| 08–0F | Local NCE |
| 10–17 | 0 |

- If entry type is LU, LU name
- If entry type is FID5 and function is associate (see HPR entry byte 07)

| Byte | Contents |
|---|---|
| 08–0B | RTP address |
| 0C–17 | 0 |

**18–1F**

- If entry type is RSREC, PCID
- If entry type is RTREC, TCID
- If entry type is LU or RTP, remote NCE
- If entry type is FID5, not applicable
- If entry type is PMI, not applicable

# HPR3 entry for HPRCTL macroinstruction (Part 3)

**Entry: HPR3**
**VIT option:**
   HPR
**Event:** Invocation of HPRCTL macroinstruction
**VIT processing module:**
   ISTITCHP

This trace record is a continuation of the HPR entry when entry type (byte 06) is RTP. It is not written, however, for some cases of FIND FIRST and FIND NEXT.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| HPR3 | 28 BYTES OF RSCV |

**Byte (hex)**
      **Contents**
**00–03**    Record ID: C"HPR3"
**04–1F**    28 bytes of RSCV

## HPR4 entry for HPRCTL macroinstruction (Part 4)

    **Entry:** **HPR4**
    **VIT option:**
        HPR
    **Event:** Invocation of HPRCTL macroinstruction
    **VIT processing module:**
        ISTITCHP

This trace record is a continuation of the HPR entry and is written when the following conditions are met:

- The HPR entry type is RTP (HPR trace record byte 6 contains R)
- CP name and FIND FIRST or FIND NEXT is specified (HPR2 trace record byte 5 has either bit 2 or bit 3 on)
- NetID or CP name (HPR2 trace record bytes 07-14) contains at least one wildcard variable
- HPRCTL is successful (HPR trace record byte 08 contains 00)

The fully qualified CP name in this record matches the CP name specified with a wildcard value in the HPR2 record.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 | 5 6 7 8 9 A B C D E F |
| H P R 4 | A 17-byte fully qualified<br>CP name | Zeros |

**Byte (hex)**
      **Contents**
**00–03**    Record ID: C"HPR4"
**04–14**    A 17-byte fully qualified CP name, left justified and padded with blanks
**15-1F**    0

## HPRT entry for HPR timer control macroinstruction

    **Entry:** **HPRT**
    **VIT option:**
        HPR
    **Event:** Invocation of HPR timer control macroinstruction

**VIT processing module:**
ISTITCHP
**Control is returned to:**
Module invoking the HPRTIMER macroinstruction

This trace record is written when the HPR timer control macroinstruction is issued. It is used to set and cancel the liveness, burst, short request, and refifo timers associated with RTP processing.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 | 1 | 1 1 1 1 1 1 | 1 1 1 1 |
|0 1 2 3|4 5 6 7|8 9 A B|C D E F|0 1 2 3|4 5|6|7 8 9 A B C|D E F|
| HPRT | A S I D / 0 / T Y P E / S E T / C A N C | TIMER DURATION | RPNCB ADDRESS | TIMER BLOCK ADDRESS | F L A G S | I N S T A N C E | ISSUING MODULE NAME | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"HPRT"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Timer type:

**B** Burst
**L** Liveness
**N** New route
**P** Path switch
**R** Refifo
**S** Short request

**07** Function:

**C** Cancel
**S** Set

**08–0B** Requested timer duration

- Burst, refifo, and short request timers, in milliseconds

- Liveness, new route, and path switch timers, in seconds

**0C–0F** RPNCB address

**10–13** Timer block address

**14-15** Timer block flags:

Byte 1

**Bit** **Meaning**
**xxx. ....**
Timer request
**000. ....**
Liveness timer
**001. ....**
Short request timer
**010. ....**
Burst timer

Appendix A. VTAM internal trace (VIT) record descriptions **219**

| **011. ....** |
| REFIFO timer |
| **100. ....** |
| HPRPST path switch timer |
| **101. ....** |
| New route for path switch timer |

**...1 ....**

Timer block is on a CLK slot.

**.... 1...**

Timer block is on the late queue.

**.... .1..**

Timer block is marked cancel.

**.... ..1.**

Timer block is on the live queue.

**.... ...1**

TPPOST of waiting RPH required for cancel function.

Byte 2

**Bit     Meaning**

**1... ....**

Timer block is on HPR timer block queue.

**.1.. ....**

Timer block is on the HPRPST queue.

**..1. ....**

Timer block is on the new route queue.

**...x xxxx**

Available.

| **16** | Instance of the trace in the issuing module. |
| **17–1B** | Name of module issuing HPRTIMER. |
| **1C–1F** | Request parameter header (RPH) address. |

# IDXx entry for identification exchange

**Entry:   IDXI or IDXO**
**VIT option:**
          CIA
**Event:**  Identification Exchange (Part 1)
**VIT processing module:**
          ISTRACCI
**Control is returned to:**
          ISTLLCXI or ISTLLCXR

This trace record contains information about inbound or outbound identification exchange processing.

| 0 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|
| IDXI IDXO | I D | 0 | CUA | NCB ADDRESS | BLOCK HEADER | 0 | RPH |

**Byte (hex)**
          **Contents**

| | |
|---|---|
| 00–03 | Record ID:<br>C"IDXI" for inbound IDX<br>C"IDXO" for outbound IDX |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05–07 | 0 |
| 08–0B | Channel Unit Address |
| 0C–0F | NCB address |
| 10–17 | Block header |
| 18–1B | 0 |
| 1C–1F | Request parameter header (RPH) address |

## IDX2 entry for identification exchange (Part 2)

**Entry: IDX2**
**VIT option:**
    CIA
**Event:** Identification Exchange (Part 2)
**VIT processing module:**
    ISTRACCI

This trace record is a continuation of the IDXx entry.

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F

  IDX2                        28 BYTES OF DATA
```

**Byte (hex)**
        **Contents**

| | |
|---|---|
| 00–03 | Record ID: C"IDX2" |
| 04–0F | 28 bytes of additional IDX data |

---

## INOP entry for the INOPINFO macro

**Entry: INOP**
**VIT option:**
    None (Generated by VTAM)
**Event:** Inoperative state
**VIT processing module:**
    ISTRACOT
**Control is returned to:**
    Module encountering inoperative

This trace record shows the invocation of the INOPINFO macro. It captures various INOP information related to a specific device.

This event is treated as an exception condition and is always traced, whether the CIO option is in effect or not.

For Local SNA Controller devices, TCP/IP DLC devices, or Channel-to-Channel devices, this INOP event is also captured within the NCB (pointed to by

NCBCIOMV).

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INOP | | | | ID | 0 | REASN | TYPE | CUA OR BLANKS | | | | NCB ADDRESS | | | | FLAGS | | | | ID | MODULE EYE CATCHER | | | | | SENSE | | STATE | 0 | | |

**Byte (hex)**

**Contents**

**01–03** Record ID: C"INOP"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Reason code

**07** Type field

> **X'41'** Connect INOP
>
> **X'42'** Signaling connection INOP
>
> **X'43'** Device INOP
>
> **X'44'** Soft INOP
>
> **X'45'** Hard INOP
>
> **X'46'** SAP INOP

**08–0B** Channel device name in EBCDIC (either a CUA or device number) or blanks

**0C–0F** NCB address

**10–13** Flags (NCBFLAGS)

**14** Product identifier. If the product identifier is S or U, then the product is VTAM. If it is V, the product is Common Storage Manager.

**15–19** Module name that detected INOP

**1A–1B** Sense code

**1C** Station state

**1D–1F** 0

# INTx entry for channel interrupt

**Entry: INTD, INTI, INTL, INTT, or INTX**

**VIT option:**
> CIO

**Event:** Channel interrupt

**VIT processing module:**
> ISTRACCI

**Control is returned to:**
> ISTTSCLC

This trace record is written when a channel program interrupt occurs.

- INTD is generated for interrupts from OSA-Express QDIO or HiperSockets adapters.
- INTI is generated for interrupts between communication controllers and local SNA cluster controllers.

- INTL is generated for interrupts from local non-SNA cluster controllers.
- INTT is generated for interrupts between TCP/IP DLC connections.
- INTX is generated for interrupts between channel-to-channel-attached hosts.

Every INT entry has a previously issued SIO entry, although it may have been issued some time ago. Use the CUA device field to correlate SIO and INT entries. The SIO entry provides additional information about this completing channel program.

Refer to *z/OS Communications Server: SNA Data Areas Volume 1* for a description of the NCB fields.

For record types with suffix D, I, X, or T, the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 | 15 | 16 17 | 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INTI<br>INTL<br>INTT<br>INTX | ID | STATE | EOPCD | DLCTYPE | CUA DEVICE | NCB ADDRESS | FLAG BYTES | FLAG | CODE | SENSE | CSW |

**Byte (hex)**
> **Contents**

**00–03**   Record ID:
> C"INTD" for DINCB
> C"INTI" for ICNCB
> C"INTL" for LDNCB
> C"INTT" for RWNCB
> C"INTX" for XCNCB

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**   For INTD, adapter state (DINCB_AFSM)

> For INTI, INTL, and INTT, link station state (NCBLNKST)

> For INTX, station state (XCNSSFSM)

**06**   For INTI, the ending operation code; otherwise, 0.

**07**   DLC type. For INTD, INTI, INTL, and INTX: 0

> For INTT:
> - For the CDLC DLC: C
> - For the LCS DLC: L
> - For the CLAW DLC: W
> - For the CTC DLC: X
> - For the HYPERchannel DLC: H

**08–0B**   Channel device name in EBCDIC (either a device address or device number)

**0C–0F**   NCB address

**10–13**   Flag bytes (NCBFLAGS)

**14**   Flag byte as follows:

| | Bit | Meaning |
|---|---|---|
| | **1... ....** | |
| | | Running in disabled interrupt exit (IOSDIE is not 0) |
| | **.1.. ....** | |
| | | ERP work area address exists (IOSERP is not 0) |
| | **..1. ....** | |
| | | Exception condition (IOSEX flag is on) |
| | **...1 ....** | |
| | | Error routine in control (IOSERR flag is on) |
| **15** | | I/O completion code (IOSCOD) |
| **16–17** | | Sense data in IOSSNS if this was a sense channel program; otherwise, 0 |
| **18–1F** | | Channel status word from IOSB |

## INT2 entry for channel interrupt (Part 2) for INTx

**Entry:** INT2
**VIT option:**
    CIO
**Event:** Channel interrupt (Part 2)
**VIT processing module:**
    ISTRACCI

This trace record is a continuation of the INTx entry.

| 0 0 0 0 | 0 0 | 0 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| INT2 | S E C P | 0 | FIRST 16 BYTES OF TRANSMIT BUFFER | 0 |

**Byte (hex)**
    **Contents**

| | |
|---|---|
| **00–03** | Record ID: C"INT2" |
| **04–05** | CPNCB_Index_Word |
| **06–07** | 0 |
| **08–17** | First 16 bytes of transmit buffer |
| **18–1F** | 0 |

---

## IO1 entry for TPIO request (Part 1)

**Entry:** IO1
**VIT option:**
    API
**Event:** Application program request (Part 1)
**VIT processing module:**
    ISTRACAP
**Control is returned to:**
    ISTAPCRS or ISTOCCSM

This trace record shows API requests. It also provides the RPL information
supplied by a VTAM application program when it issues an RPL-based
macroinstruction (such as SEND or OPNDST). The API routine ISTAICIR issues the
TPIO SVC. ISTAPCTI gets control from the SVC, does validity checking, and

queues the RPL to the PST request PAB. After the PST request PAB is dispatched, ISTAPCRS gets control and writes the entry, or calls ISTOCCSM to write the entry.

Refer to *z/OS Communications Server: SNA Programming* for an explanation of the RPL fields.

The process that is started with this event will be completed by a post (UP), or an exit (RE or UE). To correlate the two entries, match the RPL addresses.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| IO1 | | | REQTY | ID | EXTDS | OPT4 | OPT1 | RPL ADDRESS | | | | NIB ADDRESS OR CID | | | | RH3 | SRTYP | VTFL1 | VTFL2 | CHN | RU CNTL CODE | | | OPTC2 | | | | OPTC3 | | | |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"IO1"

**03** RPL request type (RPLREQ)

    **X'15'** SETLOGON
    **X'16'** SIMLOGON
    **X'17'** OPNDST
    **X'19'** CHANGE
    **X'1A'** INQUIRE
    **X'1B'** INTRPRET
    **X'1F'** CLSDST
    **X'22'** SEND
    **X'23'** RECEIVE
    **X'24'** RESETSR
    **X'25'** SESSIONC
    **X'27'** SENDCMD
    **X'28'** RCVCMD
    **X'29'** REQSESS
    **X'2A'** OPNSEC
    **X'2C'** TERMSESS

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Exit definition (RPLEXTDS)

**06** Option code byte 4 (RPLOPT4)

**07** Option code byte 1 (RPLOPT1)

**08–0B** RPL address

**0C–0F** NIB address or CID

**10** Third byte of RH (RPLRH3)

**11** Send/receive type (RPLSRTYP)

**12** VTAM flags (RPLVTFL1)

**13** Post/respond flags (RPLVTFL2)

**14** RU chain position (RPLCHN)

**15–17** RU control codes (RPLCNTRL)

**18–1B** VTAM options (RPLOPTC2)

**1C–1F** VTAM options (RPLOPTC3)

## IO2 entry for TPIO request (Part 2)

**Entry: IO2**
**VIT option:**
    API
**Event:** Application program request (Part 2)
**VIT processing module:**
    ISTRACAP

This trace record is a continuation of the IO1 entry. It shows additional information about the request.

| 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 | 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|---|
| IO2 | 0 | RETURN ADDRESS | LOGON MODE NAME OR 0 | APPLICATION LU NAME (IF AVAILABLE) | PARTNER LU NAME (IF AVAILABLE) |

**Byte (hex)**
    **Contents**
**00–02** Record ID: C"IO2"
**03** 0
**04–07** Address of the issuer of the macro
**08–0F** For SIMLOGON, OPNDST, CLSDST, or REQSESS: Logon mode name (NIBLMODE) or 0 if the NIB is not available

    For OPNSEC or TERMSESS: 0
**10–17** Application LU name if available
**18–1F** Partner LU name if available

## IO3 entry for TPIO request (Part 3)

**Entry: IO3**
**VIT option:**
    API
**Event:** Application program request (Part 3)
**VIT processing module:**
    ISTRACAP

This trace record is a continuation of the IO2 entry. It shows additional information about the request.

| 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 | 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|---|
| IO3 | 0 | ADDRESS OF USER DATA, BIND, OR 0 | TARGET LU NETID OR <br> NIBNT COUNT NRCIPBAURNMT RPOURNMT 0 | TARGET LU NAME (IF AVAILABLE) <br><br> OR 0 | PARTNER LU NETID (IF AVAILABLE) |

**Byte (hex)**

**Contents**

00–02 Record ID: C"IO3"

03 0

04–07 For SIMLOGON, CLSDST, or REQSESS: Address of user data (RPLAREA)

For OPNDST or OPNSEC: Address of BIND (NIBNDAR) or 0 if the NIB is not available

08–0F Target LU network identifier (when for CLSDST PASS)

**Byte (hex)**

**Contents**

08–09 Count of node initialization blocks (NIBs) in NIB list (when not for CLSDST PASS)

0A–0B Count of node initialization blocks (NIBs) in NIB list with NIBRPARM=0 (when not for CLSDST PASS)

0C–0F 0

10–17 Target logical unit name, if available, or 0

18–1F Partner LU network identifier, if available

---

# IPAD entry for Enterprise Extender Address Management

**Entry: IPAD**

**VIT option:**

TCP

**Event:** Enterprise Extender Address Management

**VIT processing module:**

ISTRACIP

**Control is returned to:**

The module that issued the INTRACE macroinstruction

This trace record is written when the AUNCB add, remove, find, find first, or find next function is performed.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IPAD | AS ID | 0 | FUNCTION | RC | INDEX 1 | INDEX 2 | LSAP | RSAP | IPv4 ADDRESS OR ZEROS | CALLER OF UTILITY | AUNCB ADDRESS | DATA PROVIDER ID OR ZEROS | RPH ADDRESS |

**Byte (hex)**

**Contents**

00–03 Record ID: C'IPAD'

04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05 0

06 Function:

**A** Add

**F** Find

**N** Find next

| | **R** | Remove |
|---|---|---|
| | **S** | Scan |
| | **T** | Find first |
| 07 | Return code | |
| | **X'00'** | Function succeeded |
| | **X'08'** | Function failed |
| | **X'0C'** | Function not supported |
| 08 | Index 1 | |
| 09 | Index 2 | |
| 0A | Local SAP value | |
| 0B | Remote SAP value | |
| 0C–0F | IPv4 address or 0. If 0, IPv6 address is reported in IPA2 trace record. | |
| 10–13 | Return address of the module issuing IPTADDRS | |
| 14–17 | AUNCB address | |
| 18–1B | Data provider ID or 0 | |
| 1C–1F | Request parameter header (RPH) address | |

## IPA2 entry for Enterprise Extender Address Management (Part 2)

**Entry:  IPA2**
**VIT option:**
>       TCP
**Event:**  Enterprise Extender Address Management
**VIT processing module:**
>       ISTRACIP

This trace record is a continuation of the IPAD entry. This entry will be present only when the Enterprise Extender Address Management function was performed for an IPv6 address.

| 0 0 0 0 0 1 2 3 | 0 0 0 0 0 0 0 0 0 0 0 0 4 5 6 7 8 9 A B C D E F | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|---|
| IPA2 | 0 | IPv6 ADDRESS |

**Byte (hex)**
>       **Contents**
| 00–03 | Record ID: C'IPA2' |
|---|---|
| 04–0F | 0 |
| 10–1F | IPv6 address |

## IPGN entry for host name

**Entry:  IPGN**
**VIT option:**
>       TCP
**Event:**  Host name converted to an IP address or IP address converted to a host name
**VIT processing module:**
>       ISTRACIP
**Control is returned to:**
>       The module that issued the INTRACE macroinstruction

The IPGN record is discarded whenever an Enterprise Extender host name is coverted to an IP address (either IPv4 or IPv6 address family). This record will be generated whenever a host name is converted to an IP address for the origin or destination of an EE connection.

| 0 0 0 0 0 1 2 3 | 0 4 | 0 5 | 0 0 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|---|---|
| IPGN | A S I D | 0 | LEN | IPv4 ADDRESS OR ZEROS | 0 | RETURN ADDRESS | RETURN CODE | MODULE ID | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"IPGN"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  0

**06 – 07**

IP hostname length

**08 – 0B**

IPv4 address or 0. If 0, IPv6 address is reported in IPG3 trace record.

**0C – 0F**

0

**10 –13**  Return address of the module that tried to convert the host name to an IP address.

**14–17**  Return code

**0**  Success

**1**  Host not found

**2**  Try again

**3**  No recovery

**4**  No data or no address

**18–1B**  Identifier for the module that tried to convert the host name

**1C–1F**  RPH address

# IPG2 entry for host name (Part 2)

**Entry:  IPG2**

**VIT option:**

TCP

**Event:**  Host name converted to an IP address or IP address converted to a host name

**VIT processing module:**

ISTRACIP

This trace record is a continuation of the IPGN entry. This entry may be repeated multiple times to display the entire host name.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| IPG2 | | | | IP HOST NAME CONTINUED | | | | | | | | | | | |

**Byte (hex)**
>   **Contents**
**00–03**   Record ID: C"IPG2"
**04 – 1F**
>   IP host name

## IPG3 entry for host name (Part 3)

>   **Entry:** **IPG3**
>   **VIT option:**
>   >   TCP
>   **Event:** Host name converted to an IP address or IP address converted to a host
>   >   name
>   **VIT processing module:**
>   >   ISTRACIP

This trace record is a continuation of the IPGN entry. This entry appears after the
last IPG2 record, and will be present only when the hostname resolution returns an
IPv6 address.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| IPG3 | | 0 | | | | | | IPv6 ADDRESS | | | | | | | |

**Byte (hex)**
>   **Contents**
**00–03**   Record ID: C'IPG3'
**04 – 0F**
>   0
**10 – 1F**
>   IPv6 address

## IPOG entry for getibmopt call

>   **Entry:** **IPOG**
>   **VIT option:**
>   >   TCP
>   **Event:** getibmopt is called
>   **VIT processing module:**
>   >   ISTRACIP
>   **Control is returned to:**
>   >   The module that issued INTRACE macroinstruction

This trace record is written when the getibmopt TCP/IP API function is called.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 0 0 0 0 | 1 1 | 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|0 1|2 3|4|5|6 7|8 9 A B C D E F|0 1|2 3|4 5 6 7|8 9 A B|C D E F|
| IPOG | | A S I D | 0 | NUM | NAME | S T A T U S | V E R S I O N | RETURN CODE (ERRNO) | 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"IPOG"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06–07** Number of TCP/IP jobs being reported

**08–0F** TCP/IP job name

**10–11** Status of the job (stack)

**12–13** Version of the job (stack)

**04–17** Return code (ERRNO)

**18–1B** 0

**1C–1F** Request parameter header (RPH) address

# IPO2 entry for getibmopt call (Part 2)

**Entry: IPO2**

**VIT option:**
 TCP

**Event:** getibmopt is called (Part 2)

**VIT processing module:**
 ISTRACIP

This trace record is a continuation of the IPOG entry. It contains information about the additional TCP/IP jobs.

| 0 0 | 0 0 | 0 0 0 0 0 0 0 0 | 0 0 | 0 0 | 1 1 1 1 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 |
|0 1|2 3|4 5 6 7 8 9 A B|C D|E F|0 1 2 3 4 5 6 7|8 9|A B|C D E F|
| IPO2 | | NAME1 | S T A T U S 1 | V E R S I O N 1 | NAME2 | S T A T U S 2 | V E R S I O N 2 | 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"IPO2"

**04–0B** TCP/IP job name 1

**0C–0D**

 Status of the job 1

**0E–0F** Version of the job 1

**10–17** TCP/IP job name 2

**18–19** Status of the job 2

1A–1B   Version of the job 2

## IPTC entry for altering an Enterprise Extender timer FSM state

**Entry:  IPTC**
**VIT option:**
>    TCP

**Event:**  Alteration of a timer FSM state
**VIT processing module:**
>    ISTRACIP

**Control is returned to:**
>    The module that issued the INTRACE macroinstruction

This trace record is written when the current state of an Enterprise Extender timer changes.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| IPTC | A S I D | 0 | O L D S T A T E | N E W S T A T E | TIME STAMP | NUMBER OF TIMERS ON CLOCK | NUMBER OF LIVENESS TIMERS | NUMBER OF TIMERS ON LATE QUEUE | RPH ADDRESS |

**Byte (hex)**
>    **Contents**

00–03   Record ID: C"IPTC"
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05      0
06      Old Enterprise Extender clock state

>    **P**        Pending stop

>    **R**        Running

>    **S**        Stopped

07      New Enterprise Extender clock state

>    **P**        Pending stop

>    **R**        Running

>    **S**        Stopped

08–0F   Time stamp of IPTC entry
10–13   Total number of timers currently on the Enterprise Extender clock
14–17   Number of liveness timers on the Enterprise Extender clock
18–1B   Number of timers on the late queue
1C–1D
>    Request parameter header (RPH) address

## IPTM entry for Enterprise Extender timer operation

**Entry:  IPTM**
**VIT option:**
>    TCP

**Event:** Timer set or canceled
**VIT processing module:**
    ISTRACIP
**Control is returned to:**
    The module that issued the INTRACE macroinstruction

This trace record is written when an Enterprise Extender timer is set or canceled.

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 | 1 1 | 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 | 5 6 | 7 8 9 A B | C D E F |
| IPTM | A S I D | 0 | T Y P E | S / C | DURATION OR 0 | NCB ADDRESS | TIMER BLOCK ADDRESS | F L A G S | 0 | ISSUING MODULE NAME | RPH ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"IPTM"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Timer type

   **T1** Short request

   **TL** Liveness
**07** Function

   **S** Set

   **C** Cancel
**08–0B** Request timer duration (seconds)
**0C–0F** NCB address
**10–13** Timer block address
**14** Timer block flags:

   **Bit** **Meaning**
   **xx.. ....**
      Timer type.
   **..1. ....**
      Timer block is on a CLK slot.
   **...1 ....**
      Timer block on IP timer block queue.
   **.... 1...**
      Timer block is on the late queue.
   **.... .1..**
      Timer block is marked cancel.
   **.... ..xx**
      Available.
**15–16** 0
**17–1B** Name of module setting or canceling the timer
**1C–1F** Request parameter header (RPH) address

# IRBD entry for IRB dispatch
**Entry: IRBD**

**VIT option:**
        PSS
**Event:**  IRB dispatch
**VIT processing module:**
        ISTRACPS
**Control is returned to:**
        ISTAPCIE

This trace record provides information when an interrupt request block (IRB) is dispatched. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* to determine what document contains a map of the TCB.

**Note:** For this trace record to be created, in addition to specifying the PSS option you must specify IRB or BOTH on the PSSTRACE start option.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 | 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D | E F |
| IRBD | I D | 0 | PST ADDRESS | PST SYNC TPPOSTD QUEUE | PST SYNC NORMAL QUEUE | IRB ADDRESS | TCB ADDRESS | F L G 1 | 0 |

**Byte (hex)**
        **Contents**
**00–03**   Record ID: C"IRBD"
**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07**   0
**08–0B**   PST address
**0C–0F**   PST synchronous TPPOSTed queue anchor (PSTSTPP)
**10–13**   PST synchronous normal queue anchor (PSTSNRM)
**14–17**   IRB address (TCBRBP)
**18–1B**   TCB address (PSTTCBA)
**1C–1D**
        PST flags (PSTFLG1)
**1E–1F**   0

# IRBX entry for IRB exit

**Entry:**  **IRBX**
**VIT option:**
        PSS
**Event:**  IRB exit
**VIT processing module:**
        ISTRACPS
**Control is returned to:**
        ISTAPCSD

This trace record provides information about an IRB exit.

**Note:** For this trace record to be created, in addition to specifying the PSS option you must specify IRB or BOTH on the PSSTRACE start option.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| IRBX | | | | ID | 0 | | | PST ADDRESS | | | | NEW TCB ADDRESS | | | | OLD TCB ADDRESS | | | | NEW ASCB ADDRESS | | | | OLD ASCB ADDRESS | | | | IRB ADDRESS | | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| **00–03** | Record ID: C"IRBX" |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05–07** | 0 |
| **08–0B** | PST address |
| **0C–0F** | New TCB address |
| **10–13** | Old TCB address |
| **14–17** | New ASCB address |
| **18–1B** | Old ASCB address |
| **1C–1F** | IRB address |

# IUTx entry for IUT processing (Part 1)

**Entry:** IUTC, IUTD, IUTF, IUTI, IUTM, IUTQ, IUTS, or IUTX

**VIT option:**
> CIA

**Event:** IUT processing (Part 1)

**VIT processing module:**
> ISTRACCI

**Control is returned to:**
> Module invoking the INTRACE macroinstruction that caused the record to be produced.

This trace record is written when an IUT process occurs. All CM-to-CM traffic is traced. On other sessions, only signaling traffic is traced.

## IUTC, IUTI, IUTM, IUTQ, and IUTS mapping and field descriptions

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| IUTC IUTI IUTM IUTQ IUTS | | | | ID | RESERVED | PRIMITIVE | | PROVIDER ID | | TOKEN | | STATUS | | | | TRANSACTION | | ID | | PLIST ADDRESS | | | | CALLER RETURN ADDRESS | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID:

- C"IUTC" for CONFIRM
- C"IUTI" for INDICATE
- C"IUTM" for IMMEDIATE
- C"IUTQ" for REQUEST
- C"IUTS" for RESPONSE

**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**    0

**06**    Primitive type and category

| **Bit** | **Primitive Type** |
| --- | --- |

**B'0000 ....'**
> REQUEST

**B'0001'**
> IMMEDIATE

**B'1100 ....'**
> CONFIRM

**B'0100 ....'**
> INDICATE

**B'1000 ....'**
> RESPONSE

| **Bit** | **Category** |
| --- | --- |

**B'.... 0001'**
> Connection Manager

**B'.... 0011'**
> Data Manager

**07**    Primitive Function

- When category in byte 06 is Connection Manager:

  **X'0D'**    Activate SAP

  **X'0E'**    Deactivate SAP

  **X'10'**    Call Setup

  **X'11'**    Call Clear

  **X'12'**    Call Connect

  **X'14'**    Call Status

  **X'16'**    Enable IC

  **X'18'**    Disable IC

  **X'1A'**    Call Control

- When category in byte 06 is Data Manager:

  **X'60'**    DMAct SAP

  **X'63'**    Message Unit Data

  **X'67'**    Message Data

**08–0B**    Provider ID (TOKEN)

**0C–0F**    If REQ or IND, address of PLIST list or 0. If CNF or RSP, status. Refer to *z/OS Communications Server: IP and SNA Codes* for more information about DLC status codes.

**10–13**    Transaction ID

| 14–17 | PList (Parameter List) address |
|---|---|
| 18–1B | Return address of caller |
| 1C–1F | Request parameter header (RPH) address |

# IUTD mapping and field descriptions

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IUTD | | | | ID | 0 | | PRIORITY | PROVIDER ID TOKEN | | | | STATUS | | | | CONTAINER ADDRESS (SPAC) | | | | DINCB ADDRESS | | | | CALLER RETURN ADDRESS | | | | RPH ADDRESS | | | |

**Byte (hex)**

      **Contents**

| 00–03 | Record ID: C"IUTD" for OSA-Express QDIO or HiperSockets |
|---|---|
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05–06 | 0 |
| 07 | Priority |
| 08–0B | Provider ID (TOKEN) |
| 0C–0F | Status. Refer to the *z/OS Communications Server: IP and SNA Codes* for more information about DLC status codes. |
| 10–13 | Container address (ShortPAC) |
| 14–17 | DINCB address |
| 18–1B | Return address of caller |
| 1C–1F | Request parameter header (RPH) address |

# IUTF mapping and field descriptions

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IUTF | | | | ID | 0 | SELECTOR | PROTOCOL | PROVIDER ID | | | | TOKEN | | | | EXIT ADDRESS | | | | CONTAINER ADDRESS | | | | PLIST ADDRESS | | | | CALLER RETURN ADDRESS | | | RPH ADDRESS |

**Byte (hex)**

      **Contents**

**00–03** Record ID: C"IUTF" for free PList
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Selector (see IUT2 record for selector values)
**07** Protocol (see IUT2 record for protocol values)
**08–0B** Provider ID (TOKEN)
**0C–0F** Exit address
**10–13** Container address (TIPAC)
**14–17** PList (parameter list) address
**18–1B** Caller of utility routine (caller of ISTTSCUD)
**1C–1F** Request parameter header (RPH) address

## IUTX mapping and field descriptions

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | | |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | | |
| IUTX | I D \| 0 \| F U N C T I O N | DQA Address | NCB Address | Return Address | Element Count or 0 | Called Exit Address | Thread value or 0 | | |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"IUTX" for exit call
**05** 0
**06–07** Function:
- C"WC" for write completion exit
- C"RC" for read completion exit

**08–0B** DQA address
**0C–0F** DINCB address
**10–13** INTRACE invoker
**14–17** Number of elements on queue
**18–1B** Exit address
**1C–1F** Thread value or 0

## IUT2 entry for IUT processing (Part 2)

**Entry: IUT2**
**VIT option:**
   CIA
**Event: IUT process (Part 2)**
**VIT processing module:**
   ISTRACCI

This trace record is a continuation of the IUTx trace record. It will not follow IUTD, IUTF, or IUTX records.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| IUT2 | | S E L E C T O R | P R O T O C O L | I N T E R P R E T E R | I D | USER ID TOKEN | C O N T A I N E R<br><br>(TIPAC) | A D D R E S S | RNAME (TRLE NAME) | REQUEST CORRELATOR |

**Byte (hex)**

    **Contents**

00–03    Record ID: C"IUT2"

04    Selector

    **X'00'**    RM

    **X'01'**    PTP

    **X'02'**    ATM

    **X'03'**    Samehost

    **X'81'**    CDLC

    **X'82'**    CLAW

    **X'83'**    CTC

    **X'84'**    LCS

    **X'85'**    Hyperchannel

05    Protocol

    **X'01'**    APPN

    **X'03'**    TCP/IP

    **X'04'**    UDP SAP support

    **X'05'**    X25

    **X'07'**    FastUdp

    **X'7E'**    Connection Manager

06–07    Interpreter ID (EBCDIC data)

    **ID**    **Interpreter**

    **SP**    IUTLLCSP

    **IO**    IUTLLCIO

    **II**    IUTLLCII

    **IS**    ISTALCIS

    **DP**    IUTLLCDP

    **IX**    IUTLLCIX

    **IY**    IUTLLCIY

    **SR**    IUTLLCSR

|     |     |
|-----|-----|
| **XC** | IUTLLCXC |
| **XD** | IUTLLCXD |

| | |
|-----|-----|
| **08–0B** | User ID token |
| **0C–0F** | Container address (TIPAC) |
| **10–17** | RName (TRLE Name) |
| **18–1F** | Request correlator |

## IUT3 entry for IUT processing (Part 3)

**Entry: IUT3**
**VIT option:**
    CIA
**Event:** IUT process (Part 3)
**VIT processing module:**
    ISTRACCI
**Control is returned to:**
    The module that issued the INTRACE macroinstruction

This trace record is a continuation of the IUTx trace record.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 | 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A | B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| IUT3 | LENGTH OF DATA | VERSION / SOURCE / TYPE / 0 | BUFFER TOKEN or 0 | CSM DATA SPACE ALET | ADDRESS OF DATA |

**Byte (hex)**
        **Contents**

| | |
|------|------|
| **00–03** | Record ID: C"IUT3" |
| **04–07** | Length of data |
| **08** | Version of CSM buffer descriptor |
| **09** | Buffer Source |
| | **80**     ECSA |
| | **40**     Data space |
| | **20**     User data space |
| | **10**     User storage other than a data space |
| **0A** | Buffer type |
| | **80**     Fixed |
| | **40**     Pageable |
| | **20**     Eligible to be made pageable |
| **0C–17** | Buffer token or 0 |
| **18–1B** | CSM data space ALET |
| **1C–1F** | Address of data |

## LCSx entry for LAN channel station error (Part 1)

**Entry: LCSL, LCSP, LCSS, or LCSX**
**VIT option:**
    LCS
**Event:** LAN channel station error (Part 1)
**VIT processing module:**
    ISTRACLS

**Control is returned to:**
> The module that issued the INTRACE

If the LCS trace option is specified, an LCSX trace record is created for *every* data frame VTAM receives from or sends to an IBM 3172 Interconnect Nways Controller.

If the VIT is active and VTAM receives a frame that is not valid from an IBM 3172 Interconnect Nways Controller, an LCSL, LCSP, or LCSS trace record is created depending on the following error conditions.

* The LCSL trace record is created when VTAM receives link-related data that is not valid.
* The LCSP trace record is created when VTAM receives adapter-related data that is not valid.
* The LCSS trace record is created when VTAM receives SAP-related data that is not valid.

**Note:** These records are created as an exception condition.

| 0 0 0 0 0 0 0 0 | 0 1 2 3 4 5 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|
| LCSL LCSP LCSS LCSX | I D A T D I D Y R A P P E | DEVICE | CB ADDRESS | REQUEST MODULE ABBREV NAME | DATA LENGTH | DATA ADDRESS | S T A T E | C O D E | R V I | 0 |

**Byte (hex)**
> **Contents**

**00–03** Record ID:
> C"LCSL" for information about a link
> C"LCSP" for information about an adapter
> C"LCSS" for information about an SAP
> C"LCSX" for information about a channel

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Direction

> **Code** **Meaning**
> **C"I"** Data received inbound
> **C"O"** Data sent outbound

**06** Adapter number

**07** Adapter type
> 0 = Control communication
> 2 = 802.5 (token-ring)
> 4 = CSMA/CD–802.3
> 5 = 802.4 (token-bus), MAP 3.0
> 6 = FDDI (fiber distributed data interface)

**08–0B** Channel device name in EBCDIC (either a device address or device number)

**0C–0F** Control block address
> * For LCSL, address of LSNCB
> * For LCSP, address of PRTCB

- For LCSS, address of SAPCB
- For LCSX, address of XCNCB

**10–13**  Requesting module abbreviated name
**14–17**  Data length
**18–1B**  Data address
**1C**     FSM state

For LCSL:
**00**  Nonswitched/switched reset
**01**  Nonswitched add SAP User
**02**  Nonswitched delete SAP user
**03**  Nonswitched pending deallocation
**04**  Nonswitched link active
**05**  Nonswitched open station
**06**  Nonswitched test local
**07**  Nonswitched test global
**08**  Nonswitched negotiable
**09**  Nonswitched primary
**0A**  Nonswitched connect requested
**0B**  Nonswitched pending connect indication
**0C**  Nonswitched station active
**0D**  Nonswitched close station
**81**  Switched add SAP user
**82**  Switched delete SAP user
**83**  Switched pending deallocation
**84**  Switched link active
**85**  Switched inbound answer any
**86**  Switched inbound poll pending
**87**  Switched inbound XID pending
**88**  Switched inbound pending contact
**89**  Switched inbound open station
**8A**  Switched outbound answer specific
**8B**  Switched outbound open station
**8C**  Switched outbound test local
**8D**  Switched outbound test global
**8E**  Switched outbound XID pending
**8F**  Switched outbound XID3 pending
**90**  Switched outbound pending contact
**91**  Switched pending abandon connection
**92**  Switched pending abandon connection out
**93**  Switched pending abandon connection, no station
**94**  Switched pending abandon connection out, no station
**95**  Switched close station
**96**  Switched answer off
**97**  Switched negotiable 1
**98**  Switched negotiable 2
**99**  Switched negotiable 3
**9A**  Switched primary
**9B**  Switched secondary
**9C**  Switched connect requested
**9D**  Switched pending connect indication
**9E**  Switched station active
**9F**  Switched inbound XID negotiation-preceding pending

For LCSP:
**00**  Reset

| | | |
|---|---|---|
| | **01** | Add channel user pending |
| | **02** | Adapter enable pending |
| | **03** | Active |
| | **04** | Adapter disable pending |
| | **05** | Delete channel user pending |
| | **06** | Inoperative |
| | **07** | Blocked |

For LCSS:

| | | |
|---|---|---|
| | **00** | Reset |
| | **01** | Add port user pending |
| | **02** | Add SRM entry pending |
| | **03** | Activate SAP pending |
| | **04** | Active |
| | **05** | Deactivate SAP pending |
| | **06** | Delete SRM entry pending |
| | **07** | Delete port user pending |
| | **08** | Inoperative |
| | **09** | Blocked |

For LCSX:

| | | |
|---|---|---|
| | **00** | Reset state |
| | **01** | X-side I/O pending |
| | **02** | Y-SIDE I/O pending |
| | **03** | Contact wait |
| | **04** | Contact wait DCM |
| | **05** | XID7 1 pending |
| | **06** | XID7 2 pending |
| | **07** | XID bad pending |
| | **08** | XID7 1 I/O pending |
| | **09** | XID7 2 I/O pending |
| | **0A** | XID bad I/O pending |
| | **0B** | Ready |
| | **0C** | Discontact scheduled |
| | **0D** | HDV pending |
| | **0E** | HDV pending DCM |
| | **0F** | Discontact pending |
| **1D** | Reason code | |
| | **00** | Successful. |
| | **04** | Exit PAB immediately. |
| | **08** | Entire primitive length is not valid. |
| | **0C** | Buffer allocation failed. |
| | **10** | Length of primitive header is not valid. |
| | **14** | Length of data field is not valid. |
| | **18** | Incorrect target layer identifier. |
| | **1C** | Incorrect identifier type. |
| | **24** | Correlator ID mismatch. |
| | **28** | Incorrect parameter list version. |
| | **2C** | Primitive was received in wrong state. |
| | **30** | Unrecognized primitive code. |
| | **34** | Incomplete primitive header. |
| | **38** | User_sap_id provided is not valid. |
| | **58** | Length of routing vector (if included) is not valid. |
| | **5C** | XID command/response value is not valid. |
| | **60** | Poll/final indicator is not valid. |
| | **64** | Unexpected XID type. |

| | |
|---|---|
| **68** | Negative confirm received. |
| **6C** | User class is not valid. |
| **70** | Flow action value is not valid. |
| **74** | SAP is not net manager capable. |
| **78** | Not valid for net manager SAP. |
| **7C** | Vector length field is not valid in adapter-specific overlay. |
| **80** | Length of adapter-specific overlay field is not valid. |
| **84** | Vector identifier field is not valid. |
| **88** | Destination MAC/Destination SAP reporting field is not valid. |
| **8C** | Identifier (user_sap_id or user_cep_id) is inconsistent with identifier type. |
| **94** | Not valid for logon type. |
| **98** | Test/XID remote MAC/remote SAP does not match an entry in the SAP user matrix (SUM), and SUM has no "answer any" entries available. |
| **9C** | Pacing fields are not 0. |
| **A0** | Success count not X'FF'. |
| **A4** | user_cep_id value not in SAP user matrix (SUM) range or SUM entry not in use or answer mode not specific for SUM entry. |
| **A8** | Adapter is inactive. |
| **AC** | No match in adapter routing table found or incorrect LAN type. |
| **B0** | XCNCB 8 byte header error. |
| **B4** | XCNCB byte count mismatch. |
| **B8** | XCNCB bad primitive offset. |

| | | |
|---|---|---|
| **1E** | RVI | |
| | | C"N" = route vector not included |
| | | C"Y" = route vector included |
| **1F** | 0 | |

# LCS2 entry for LAN channel station error (Part 2)

**Entry: LCS2**
**VIT option:**
> LCS

**Event:** LAN channel station error (Part 2)
**VIT processing module:**
> ISTRACLS

This trace record is a continuation of the LCSL, LCSP, LCSS, or LCSX entry.

The VTAM internal trace will generate as many as 31 LCS2 records for the LCSL, LCSP, LCSS, and LCSX entries.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| LCS2 | UP TO 28 BYTES OF CHANNEL DATA |

**Byte (hex)**
> **Contents**

| | |
|---|---|
| **00–03** | Record ID: C"LCS2" |
| **04–1F** | Up to 28 bytes of channel data |

# LCSM entry for MACaddress for DATA SEND/RECEIVE

**Entry: LCSM**
**VIT option:**
  LCS
**Event:** MACaddress for DATA SEND/RECEIVE
**VIT processing module:**
  ISTRACLS

If the LCS trace option is specified, an LCSM trace record is created for *every* data frame VTAM sends to and receives from an IBM 3172 Interconnect Nways Controller.

The LCSM trace record is created in association with an LCSX entry. However, because these trace entries are created by different component elements, the entries may be separated in the trace by other component traces. For outbound data, the LCSM entry will appear before the LCSX entry. For inbound data, the LCSM entry will appear following the LCSX entry.

| 0 0 0 0 | 0 | 0 | 0 0 0 0 0 0 | 0 0 | 0 0 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 8 9 A B | C D | E F 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| LCSM | I D | D I R | DESTINATION MACADDRESS | L A N O P R | SOURCE MACADDRESS | INDEX 1 | INDEX 2 | R P H    A D D R E S S |

**Byte (hex)**
  **Contents**
**00–03**   Record ID: C"LCSM" for information about macaddress
**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**    Direction

  **Code**   **Meaning**
  **C"I"**    Data received inbound
  **C"O"**    Data sent outbound
**06-0B**   Destination macaddress
**0C-0D**   LAN Operation

  **Note:** It is not required that users of the product know the meaning of this internal VTAM name. When required, the product support organization may use it to assist in internal flow diagnosis.
**0E–13**   Source MACaddress
**14–17**   Index 1

  **Note:** It is not required that users of the product know the meaning of this internal VTAM name. When required, the product support organization may use it to assist in internal flow diagnosis.
**18–1B**   Index 2

> **Note:** It is not required that users of the product know the meaning of this internal VTAM name. When required, the product support organization may use it to assist in internal flow diagnosis.
>
> **1C–1F** Request parameter header (RPH) address

---

# LDLC entry for invoking LDLC

**Entry: LDLC**
**VIT option:**
    SSCP
**Event:** Invocation of an LDLC module
**VIT processing module:**
    ISTRACSC
**Control is returned to:**
    The module that issued the INTRACE macroinstruction

This trace record is issued when an LDLC module is invoked.

| 0 0 0 0<br>0 1 2 3 | 0<br>4 | 0<br>5 | 0<br>6 | 0<br>7 | 0<br>8 | 0 0 0 0 0 0 0<br>9 A B C D E F | 1 1 1 1<br>0 1 2 3 | 1 1 1 1<br>4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| LDLC | A S I D | 0 | D L C T Y P E | D E S T I N A T I O N | C O D E | 0 | RETURN ADDRESS | WORK ELEMENT ADDRESS | MODULE NAME | AUCPL ADDRESS |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"LDLC"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Data length control type:

    **A**      ATM

    **E**      Enterprise Extender
**07** Destination
**08** Code
**09–0F** 0
**10–13** Return address of the module that invoked the LDLC
**14–17** Work element address
**18–1B** Name of the LDLC module
**1C–1F** Address of the ALCPL or AUCPL

## LDL2 entry for invoking LDLC (Part 2)

**Entry: LDL2**
**VIT option:**
    SSCP
**Event:** Invocation of an LDLC module Part 2

This trace record is a continuation of an LDLC module. It shows the variable data portion of the control block.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| LDL2 | 24 BYTES OF DATA | 0 |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"LDL2"
**04–1B** Data
**1C–1F** 0

## LKEX entry for TPLOCK exclusive

**Entry: LKEX**
**VIT option:**
    LOCK
**Event:** TPLOCK exclusive
**VIT processing module:**
    ISTRACLK
**Control is returned to:**
    ISTAPC36

This trace record identifies a request for exclusive control of a VTAM lock and the current status of a lock. Exclusive locks can be held by only one VTAM process. If a WAIT entry immediately follows this entry, it *could* be because a wait-for-lock condition exists. Look at the lockword in the trace output to see if the lock is already held.

Locks obtained by LKEX are eventually released by UNLK or ULKA. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for a list of VTAM locks.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| LKEX | I D | 0 | L O C K | L E V E L | LOCK ADDRESS | CRA LKACT | RETURN ADDRESS | LOCKWORD | L I D | RPH ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"LKEX"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06–07** Level of lock to be obtained

08–0B    Address of lock to be obtained
0C–0F    CRA lock account word (CRALKACT)
10–13    Address of the issuer of the TPLOCK macro
14–1A    Lockword pointed to by address in bytes 08-0B
1B          The LOCK ID field. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for a list of VTAM locks.
1C–1F    Request parameter header (RPH) address

# LKSH entry for TPLOCK shared

**Entry:  LKSH**
**VIT option:**
    LOCK
**Event:**  TPLOCK shared
**VIT processing module:**
    ISTRACLK
**Control is returned to:**
    ISTAPC35

This trace record identifies a request for a shared VTAM lock and the current status of a lock. Shared locks can be held by more than one VTAM process at a time. If a WAIT entry immediately follows this entry, it may be because a wait-for-lock condition exists. Look at the lockword in the trace output to see if the lock is already held.

Locks obtained by LKEH are eventually released by UNLK or ULKA. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for a list of VTAM locks.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 18 19 1A | 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| LKSH | ID | STATUS | LOCK | LEVEL | LOCK ADDRESS | CRA LKACT | RETURN ADDRESS | LOCKWORD | LID | RPH ADDRESS |

**Byte (hex)**
    **Contents**
00–03    Record ID: C"LKSH"
04          ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05          Indicates lock wait status:
    **X'00'**    WAIT
    **X'80'**    NOWAIT
06–07    Level of lock to be obtained
08–0B    Address of lock to be obtained
0C–0F    CRA lock account word (CRALKACT)
10–13    Address of the issuer of the TPLOCK macro
14–1A    Lockword pointed to by address in bytes 08-0B
1B          The LOCK ID field. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for a list of VTAM locks.
1C–1F    Request parameter header (RPH) address

# LNKx entry

**Entry:** LNKA, LNKI, LNKL, LNKU, or LNKX
**VIT option:**
    CIA
**Event:** Change to link use count
**VIT processing module:**
    ISTRACCI
**Control is returned to:**
    The module that issued the INTRACE macroinstruction

This trace record is written when the link use count in the node control block (NCB) is incremented or decremented.

| 0 0 0 0 | 0 4 | 0 5 | 0 6 0 7 | 0 8 0 9 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C | 1D | 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LNKA LNKI LNKL LNKU LNKX | ID | REASON | ELEMENT ADDRESS | LINK USE COUNT | NCB ADDRESS | ADDRESS OF CONTROL BLOCK | 0 | MODULE OR CALLER ADDRESS | NCB | DEPENDENT | DATA |

**Byte (hex)**

**Contents**

**00–02**  Record ID: C"LNK"

**03**  NCB type:
    C"LNKA" for AHNCB
    C"LNKI" for ICNCB
    C"LNKL" for LDNCB
    C"LNKR" for RPNCB
    C"LNKU" for AUNCB
    C"LNKX" for XCNCB

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  Reason for change to use count (DNPFUNC):
    **X'01'**  Increment for boundary SLU element address (BLB)
    **X'02'**  Increment for boundary PLU element address (BPAT)
    **X'03'**  Increment for boundary SNA half-session (BSB)
    **X'04'**  Increment for boundary non-SNA half-session (LUST)
    **X'05'**  Increment for application half-session (FMCB)
    **X'06'**  Increment for rapid transport protocol (RTP) half-session queue
    **X'07'**  Increment for rapid-transport protocol (RTP)
    **X'F1'**  Decrement for boundary SLU element address (BLB)
    **X'F2'**  Decrement for boundary PLU element address (BPAT)
    **X'F3'**  Decrement for boundary SNA half-session (BSB)
    **X'F4'**  Decrement for boundary non-SNA half-session (LUST)
    **X'F5'**  Decrement for application half-session (FMCB)
    **X'F6'**  Decrement for line (NCB)
    **X'F7'**  Decrement for rapid-transport protocol (RTP) half-session queue
    **X'F8'**  Decrement for rapid-transport protocol (RTP)

**06–07**  Element address (for boundary element address reasons)

**08–0B**  Link use count after the increment or decrement

**0C–0F**  Node control block (NCB) address

**10–13**  Address of control block associated with the increment or decrement

14–15    0
16–17    Element index value
18–1B    Module address or address of the issuer of the TSCDN caller
1C–1F    NCB Dependent Data

> **For LNKU entries:**
> > Represents the number of non-RS (route setup) RTP pipes that originate in this host and traverse this Enterprise Extender connection.
>
> **For all other entries:**
> > 0

## LOST entry for lost trace record

**Entry:  LOST**
**VIT option:**
> None

**Event:  Lost trace record**
**VIT processing module:**
> ISTRACRR

**Control is returned to:**
> The location shown in register 14

This trace record indicates that the internal trace table is incomplete. Information from events that were supposed to be traced was lost because there were not enough save areas available for the tracing routine to record the information.

| 0 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| LOST | I D | 0 | REG 14 | | 'LOST TRACE RECORD' | | | |

**Byte (hex)**
> **Contents**
00–03    Record ID: C"LOST"
04       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05–07    0
08–0B    Register 14 (return address)
0C–1F    C"LOST TRACE RECORD"

## LSNA entry for local SNA record (Part 1)

**Entry:  LSNA**
**VIT option:**
> CIA

**Event:  Read/Write from APPN PU Channel**
**VIT processing module:**
> ISTRACCI

**Control is returned to:**
> ISTTSC86

This trace record is written each time data is read or written across an APPN host-to-host channel.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| LSNA | | I D | C B I D | F L A G S | T Y P E | TSCB ADDRESS | DAT LEN | 18 BYTES OF LOCAL SNA DATA | | | | | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"LSNA"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID of work element being written or read across the channel

**06** Event Flags

**Bit** **Meaning**

**1... ....**
The data unit being traced (FID2 data or control data) is the last data unit in the multipath channel (MPC) transmit block header just received. This bit only has meaning for inbound data.

**.1.. ....**
This data is control data. If off, this data is normal FID2 data traffic.

**..1. ....**
This trace entry contains the beginning of a new PIU/control data. If off, this data unit is the middle or end portion of a PIU. For outbound data, this bit will always be on (only the first part of PIU/Control data outbound is traced).

**07** A 1-byte field that indicates the direction of the data unit flow:

**Code** **Meaning**
C"I" Data received inbound
C"O" Data sent outbound

**08–0B** TSCB address

**0C–0D**
Length of data being sent or received

**0E–1F** This 18-byte field will map as follows:

- For control data
  - The 2-byte QLLC Header (indicating the type of the control data)
  - Up to 16 bytes of the remaining control data

- For FID2 data
  - The 6-byte FID2 TH header
  - A 3-byte RH if it exists
  - Up to 9 bytes of the remaining FID2 data (12 bytes if no RH)

## LSN2 entry for local SNA record (Part 2)

**Entry: LSN2**

**LSN2**

**VIT option:**
> CIA

**Event:** Read/Write from APPN Host PU Channel

**VIT processing module:**
> ISTRACCI

This trace record is a continuation of the LSNA entry. It contains 28 more bytes of data.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| LSN2 | 28 MORE BYTES OF LOCAL SNA DATA | |

**Byte (hex)**
> **Contents**

00–03    Record ID: C"LSN2"

04–1F    Up to 28 more bytes of data

# MCO1 entry for MIBConnect (Part 1)

**Entry:** MCO1

**VIT option:**
> CMIP

**Event:** MIBConnect called by CMIP application program

**VIT processing module:**
> ISTITCCM

**Control is returned to:**
> Module invoking the INTRACE macro that caused the record to be produced

This trace record is generated when a CMIP application program calls the MIBConnect function. The trace record shows the information that is passed from the application program to the MIB controller and from the MIB controller to the application program.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

If either the MIBConnect or OPEN ACB return code is not 0, this entry is always traced, regardless of the VIT options specified.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| MCO1 | I D | A C B R C | R C | LINK ID | MAXIMUM OUT-STANDING | APPLICATION NAME OR BLANKS | 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"MCO1"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code from OPEN ACB

**06–07** MIBConnect return code

**08–0B** Link identifier

**0C–0F** Maximum number of outstanding invoke identifiers

**10–17** Application name (padded on the right with blanks) or blanks. This field is blanks if the MIBConnect module cannot obtain storage for local variables.

**18–1B** 0

**1C–1F** Request parameter header (RPH) address

# MCO2 entry for MIBConnect (Part 2)

**Entry:** MCO2

**VIT option:**
CMIP

**Event:** MIBConnect called by CMIP application program (Part 2)

**VIT processing module:**
ISTITCCM

This trace record is a continuation of the MCO1 entry and contains additional information that is passed from the application program to the MIB controller and from the MIB controller to the application program.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

| 00 01 02 03 | 04 05 | 06 07 | 08 | 09 | 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| MCO2 | SMA | SYS OBJ | L E V E L | I D L E N | 0 | TERMIN EXIT ADDRESS OR 0 | READ-Q EXIT ADDRESS OR 0 | USER DATA | DATA SPACE NAME |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"MCO2"

**04–05** System management application entity (SMAE) name length on input. This field is X'FFFF' if the value is greater than 65 635.

**06–07** System object name length on input. This field is X'FFFF' if the value is greater than 65 635.

**08** API level. This field is X'FF' if the value is greater than 255.

**09** Local identifier field. This field is X'FF' if the value is greater than 255.

**0A–0B** 0

**0C–0F** Termination exit address or 0. If nonzero, this TPEND exit is driven and this field can be used to correlate with the UE1 and UE2 VIT entries.

**10–13** Read-queue exit address or 0

**14–17** User data

**18–1F** Data space name, or blanks (if no data space name is provided)

# MDEL entry for MIBSendDeleteRegistration

**Entry: MDEL**

**VIT option:**
CMIP

**Event:** MIBSendDeleteRegistration called by CMIP application program

**VIT processing module:**
ISTITCCM

**Control is returned to:**
Module invoking the INTRACE macro that caused the record to be produced

This trace record is generated when a CMIP application program calls the MIBSendDeleteRegistration function of the CMIP services API to delete the registration of an object. The trace record shows the information that is passed from the application program to the MIB controller and from the MIB controller to the application program.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

If the return code is not 0, this entry is always traced, regardless of the VIT options specified.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D | E F |
| MDEL | I D | 0 | R C | LINK ID | INVOKE ID | LOCAL ID OR 0 | DN ADDRESS OR 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"MDEL"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06–07** MIBSendDeleteRegistration return code

**08–0B** Link identifier

**0C–0F** Invoke identifier. The invoke identifier can be used to correlate this VIT entry with the CMIP message string in the buffer trace.

**10–17** Local identifier (padded on the right with zeros) or 0

**18–1B** Address of distinguished name or 0

**1C–1F** Request parameter header (RPH) address

# MDIS entry for MIBDisconnect

**Entry: MDIS**

**VIT option:**
CMIP

**Event:** MIBDisconnect called by CMIP application program

**VIT processing module:**
ISTITCCM

**Control is returned to:**
>    Module invoking the INTRACE macro that caused the record to be
>    produced

This trace record is generated when a CMIP application program calls the
MIBDisconnect function of the CMIP services API. The trace record shows the
information that is passed from the application program to the MIB controller and
from the MIB controller to the application program.

**Note:** If the CMIP application program is internal, such as the VTAM topology
>    agent, the MIBDisconnect function is not traced.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP
Services and Topology Agent Guide*.

If either the MIBDisconnect or CLOSE ACB return code is not 0, this entry is
always traced, regardless of the VIT options specified.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| MDIS | | 0 | A C B R C | R C | LINK ID | 0 | | | | |

**Byte (hex)**
>    **Contents**

**00–03**  Record ID: C"MDIS"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is
>    greater than X'FF'.
**05**  Return code from CLOSE ACB
**06–07**  MIBDisconnect return code
**08–0B**  Link identifier
**0C–1F**  0

# MMG entry for memory management (Part 1)

**Entry:  MMG**
**VIT option:**
>    None
**Event:**  ISTORMMG detected an error condition.
**VIT processing module:**
>    ISTRACTR
**Control is returned to:**
>    ISTORMMG

This entry is written when ISTORMMG detects an overlay in the storage obtain or
to be freed queue.

| 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| MMG | F U N C C | FREEPTR | TO_BE_ FREED | HEADER_ PTR | The contents of SMHDR pointed by HEADER_PTR for x'10' bytes |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"MMG"

**03** Function code:

**1** Running through the single threaded storage obtained queue. The storage management header has invalid information pointed to by FREEPTR.

**2** Running through the local TO_BE_FREED queue. The storage management header pointed to by FREEPTR contains invalid information. FREEPTR was copied from TO_BE_FREED.

**3** Running through the local TO_BE_FREED queue. The backward pointer has invalid information in its storage management header.

**4** Running through the local TO_BE_FREED queue. The forward pointer has invalid information in its storage management header.

**04–07** FREEPTR (Address of the storage to be freed)

**08–0B** Local TO_BE_FREED queue pointer

**0C-0F** HEADER_PTR ( The current storage management header address)

**10–1F** The contents of SMHDR pointed to by HEADER_PTR

## MM2 entry for memory management (Part 2)

**Entry: MM2**

**VIT option:**

None

**Event:** Continuation of MMG trace entry

**VIT processing module:**

ISTRACTR

| 0 0 0 | 0 | 0 0 0 0 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 7 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| MM2 | F U N C C | 0 | FREEPTR | The contents of SMHDR pointed by FREEPTR for x'10' bytes OR ZERO |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"MM2"

**03** Function code:

**1** Running through the single threaded storage obtained queue. The storage management header has invalid information pointed to by FREEPTR.

**2** Running through the local TO_BE_FREED queue. The storage management header pointed to by FREEPTR contains invalid information. FREEPTR was copied from TO_BE_FREED.

> **3** Running through the local TO_BE_FREED queue. The backward pointer has invalid information in its storage management header.
>
> **4** Running through the local TO_BE_FREED queue. The forward pointer has invalid information in its storage management header.

**04–0B** 0
**0C-0F** FREEPTR (Address of the storage to be freed)
**10–1F** The contents of SMHDR pointed to by FREEPTR

## MNPS entry for MNPS macro

**Entry: MNPS**
**VIT option:**
  CFS
**Event:** Multinode persistent session coupling facility access
**VIT processing module:**
  ISTRACCF
**Control is returned to:**
  Module invoking the coupling facility access that caused the record to be produced

This entry is written when VTAM manipulates data in the MNPS coupling facility structure.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| MNPS | ID | 0 | FUNCTION | DATA TYPE | MNPS APPLICATION NAME | RETURN ADDRESS | RETURN CODE | ACCESS TOKEN | RPH ADDRESS |

**Byte (hex)**
  **Contents**
**00–03** Record ID: C"MNPS"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Function request by this invocation of MNPS:
  **X'01'** Read
  **X'02'** Update
  **X'03'** Delete
**07** Type of data being updated:
  **X'01'** Application status block (ASB)
  **X'02'** FMCB
  **X'03'** FMCB extension
  **X'04'** BSB
  **X'05'** Inbound CV29
  **X'06'** Outbound CV29
  **X'07'** RTP
  **X'08'** NLP
  **X'09'** LM Table

**MNPS**

X'0A'   SAB
X'0B'   HSICB
08–0F   Name of the multinode persistent session application program
10–13   Address of the invoker of the MNPS macro
14–17   Return code from the MNPS macro
18–1B   CFS access token
1C–1F   Request parameter header (RPH) address

# MPDU entry for MPC PDU processing

**Entry: MPDU**
**VIT option:**
        CIA
**Event:** Inbound or outbound PDU over high performance data transfer
**VIT processing module:**
        ISTRACCI
**Control is returned to:**
        ISTTSCDP and IUTLLCMB for inbound processing or ISTTSC88 for
        outbound processing

The MPDU trace record shows outbound and inbound MPC PDU processing of
CSM buffers.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 | 0 | 0 | 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A | B | C | D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| MPDU | I D | 0 | ML | TLN | F | I | ACT | PDU SEQ | XBUFLST POINTER | PDU HEADER POINTER | RPH POINTER |

**Byte (hex)**
        **Contents**
00–03   Record ID: C"MPDU"
04      ID is the primary address space ID (ASID). This field is 0 if the ASID is
        greater than X'FF'.
05      0
06–07   PDU moved data length
08–0A   PDU total data length
0B      PDU header flag byte (See ISTPDHDR - PDHFLAGS)
0C      Protocol ID (See ISTPDHDR - PDHPID)
0D–0F   PDU route action word
        **OSD**   Outbound data
        **IRS**   A PDU is routed to MPC DLC user.
        **IRG**   A PDU is routed to MPC DLC user with queued PDUs.
        **ICF**   The first M2A wait entry is created with the PDU.
        **ICA**   A PDU is queued to a new M2A entry. The new M2A entry has a
                backward chain to an existing M2A entry.
        **ICB**   A PDU is queued to a new M2A entry. The new M2A entry has a
                forward chain to an existing M2A entry.
        **IEE**   Enterprise extender input.
        **IQH**   A PDU is queued to a head of the M2A entry.
        **IQT**   A PDU is queued to a tail of the M2A entry.
        **IQC**   A PDU is queued to M2A entry. Two M2A entries are combined
                into one M2A.

**IDP**    A duplicate PDU is discarded.
**ICT**    Connection termination is requested.
**OEE**   Enterprise extender output.
**10–13**  PDU sequence number or X'00000000' for CNLS PDU
**14–17**  Address of the first ISTXBFLST
**18–1B**  Address of the PDU header
**1C–1F**  Address of VTAM RPH

## MPD2 entry for MPC PDU processing

**Entry: MPD2**
**VIT option:**
    CIA
**Event:** Inbound or outbound PDU over high performance data transfer
**VIT processing module:**
    ISTRACCI

This trace record is a continuation of the MPDU trace record. Up to three MPD2 trace records follow an MPDU trace record, each containing 28 bytes of data from the extended buffer list (XBUFLST) associated with this PDU.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| MPD2 | 28 BYTES FROM XBUFLST |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"MPD2"
**04–1F**  28 bytes of data from the extended buffer list (XBUFLST) associated with this PDU

## MQRQ or MQRS entry for MIBSendRequest or MIBSendResponse

**Entry: MQRQ** or **MQRS**
**VIT option:**
    CMIP
**Event:** MIBSendRequest or MIBSendResponse called by CMIP application
    program
**VIT processing module:**
    ISTITCCM
**Control is returned to:**
    Module invoking the INTRACE macro that caused the record to be
    produced

The MQRQ is generated when a CMIP application program calls the MIBSendRequest function of the CMIP services API to send a request. The MQRS is generated when a CMIP application program calls the MIBSendResponse function of the CMIP services API to send a response. The trace record shows the information that is passed from the application program to the MIB controller and from the MIB controller to the application program.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

If the return code is not 0, this entry is always traced, regardless of the VIT options specified.

**Note:** Calls to MIBSendCmipRequest and MIBSendCmipResponse functions also generate these trace records, but there is not a one-to-one correspondence.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|
| MQRQ MQRS | I D | 0 | R C | LINK ID | INVOKE ID | LOCAL ID OR 0 | MESSAGE ADDRESS | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID:
 C"MQRQ" for MIBSendRequest
 C"MQRS" for MIBSendResponse

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06–07** MIBSendRequest or MIBSendResponse return code

**08–0B** Link identifier

**0C–0F** Invoke identifier. The invoke identifier can be used to correlate this VIT entry with the CMIP message string in the buffer trace.

**10–17** Local identifier (padded on the right with zeros) or 0

**18–1B** Address of the message

**1C–1F** Request for parameter header (RPH) address

# MREG entry for MIBSendRegister

**Entry:** **MREG**
**VIT option:**
 CMIP
**Event:** MIBSendRegister called by CMIP application program
**VIT processing module:**
 ISTITCCM
**Control is returned to:**
 Module invoking the INTRACE macro that caused the record to be produced

This trace record is generated when a CMIP application program calls the MIBSendRegister function of the CMIP services API to register an object. The trace record shows the information that is passed from the application program to the MIB controller and from the MIB controller to the application program.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

If the return code is not 0, this entry is always traced, regardless of the VIT options specified.

| 0 0 0 0 | 0 4 | 0 5 | 0 6 0 7 | 0 8 0 9 0 A 0 B | 0 C 0 D 0 E 0 F | 1 0 1 1 1 2 1 3 1 4 1 5 1 6 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C 1 D 1 E 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MREG | I D | F L A G S | R C | LINK ID | INVOKE ID | LOCAL ID | T Y P E | A L L O | C R E A T | 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"MREG"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Capability flags. This field is X'FF' if the value is greater than 255.

**Bit     Meaning**

**.... ..1.**

Reserved

**.... ...1**

Subtree handler

**06–07** MIBSendRegister return code

**08–0B** Link identifier

**0C–0F** Invoke identifier. The invoke identifier can be used to correlate this VIT entry with the CMIP message string in the buffer trace.

**10–17** Local identifier (padded on the right with zeros)

**18** Name type. This field is X'FF' if the value is greater than 255.

**19** Allomorphs count. This field is X'FF' if the value is greater than 255.

**1A** Create handlers count. This field is X'FF' if the value is greater than 255.

**1B** 0

**1C–1F** Request parameter header (RPH) address

# MRG entry for updates to the VTAM topology agent (Part 1)

**Entry: MRG**

**VIT option:**

CMIP

**Event:** Resource updates sent to VTAM topology agent

**VIT processing module:**

ISTITCCM

**Control is returned to:**

Module invoking the INTRACE macro that caused the record to be produced

This trace record is generated when the VTAM topology agent receives updates from VTAM for changes in VTAM resources.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| MRG | | | OBJ TYPE | ID | 0 | FLAGS | 0 | NAME OF RESOURCE CAUSING THE INCOMING UPDATE | | | | | | | | I REASON | P REASON | I AVC | P AVC | I OBJ PRE | P OBJ PRE | NUM MERG | WORK AREA | TYPE REQ | 0 | RPH ADDRESS | | | | | |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"MRG"

**03** Object type. The type of resource causing the incoming update.

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Action performed because of the incoming update.

**Bit**    **Meaning**

**1... ....**

The incoming update is waiting to be merged with older updates before it is sent in a CMIP message.

**.1.. ....**

The incoming update is merged with older updates for resources that own the resource that caused this incoming update.

**..1. ....**

The incoming update is merged with older updates for the same resource that caused the incoming update.

**...1 ....**

The resource causing the incoming update has a different vertex 1 from the older, existing update for the same resource. This bit has meaning only when bit 2 is on.

**.... 1...**

The vertex 1 or line of the older, existing update is the resource that caused the incoming update. This bit has meaning only when bit 3 is on.

**.... .1..**

An older, existing update is sent before the incoming update because the incoming update cannot be merged with the older update.

**.... ..00**

Event report or notification

**.... ..01**

SNAlocaltopology

**.... ..10**

LUcollection

**.... ..11**

SNAnetwork

**07** 0

**08–0F** Name of the resource causing the incoming update

**10** Reason for the incoming update

**11** Reason for the older, existing update, if there is one

**12** Change in attribute value caused by the incoming update
**13** Change in attribute value caused by the older, existing update, if there is one
**14–15** Object flags indicated on the incoming update
**16–17** Object flags indicated on the older, existing update, if there is one
**18** The number of times resource updates for this particular resource were merged before the incoming update is processed. This number does not include the merges that occurred for resources that own this particular resource.
**19** Workarea flags
**1A** Flags contained in the Agent Support Request Mapping control block
**1B** 0
**1C–1F** Request parameter header (RPH) address

# MRG2 entry for updates to the VTAM topology agent (Part 2)

**Entry: MRG2**
**VIT option:**
    CMIP
**Event:** Resource updates sent to VTAM topology agent
**VIT processing module:**
    ISTITCCM

This trace record is a continuation of the MRG entry.

| 0 0 0 0<br>0 1 2 3 | 0 0 0 0<br>4 5 6 7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1<br>0 1 | 1 1 1 1 1 1<br>2 3 4 5 6 7 | 1 1 1 1 1 1 1 1<br>8 9 A B C D E F |
|---|---|---|---|---|---|---|
| MRG2 | RES STATE ON INCOMIN UPDATE | RES STATE ON EXISTIN UPDATE | OLDEST RES STATE | O B J C O U N T | 0 | NAME OF OWNING RESOURCE |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"MRG2"
**04–07** New state of the resource that caused the incoming update
**08–0B** New state of the resource that caused the incoming update, as indicated on the older, existing update, if there is one
**0C–0F** Oldest state of the resource causing the incoming update. If there is an older, existing update, this is the old state specified on the update that is waiting. Otherwise, this is the old state of the resource causing the incoming update.
**10–11** Object count on the incoming update
**12–17** 0
**18–1F** Value depends on type of update:

    **Type of Update**
        **Value**
    **Event report or notification**
        0

> **SNAnetwork**
> > Name of the vertex 1
>
> **LUcollection**
> > Name of the PU for which LUcollection is requested
>
> **SNAlocaltopology**
> > Name of the vertex 1. If the incoming update is merged with an older, existing update for a resource that owns the resource causing the incoming update, this is the name of the owning resource.

# MSG entry for message issued

**Entry: MSG**
**VIT option:**
> MSG

**Event:** Message issued (Part 1)
**VIT processing module:**
> ISTRACOT

**Control is returned to:**
> ISTCFCTM

This trace record is written each time a message is issued by a VTAM module. You can use it to find the originator of a message and to see whether the message was solicited or unsolicited.

MSG trace entries are generated for all messages, even those that are suppressed by message-flooding prevention. MSG trace entries thus provide information that may be missing from the operator's console.

Refer to *z/OS Communications Server: SNA Messages* for information on the message flooding prevention. For information on the MODIFY SUPP command, refer to *z/OS Communications Server: SNA Operation*. For information on the SUPP start option, refer to *z/OS Communications Server: SNA Resource Definition Reference*.

The VTAM Internal Trace (VIT) does not trace the following messages:
- Logon manager messages (beginning with ELM)
- TSO/VTAM messages (beginning with IKT)

| 00 01 02 03 | 04 05 | 06 | 07 08 09 0A | 0B 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|
| MSG | I D | 0 | S O R U | MSG ID OR 'USS' | MODULE ID | SAVE AREA ADDRESS | RETURN ADDRESS | DESTINATION OR 0 |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"MSG"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0

**06** 'S' if the message was solicited (issued in response to a VTAM command); 'U' if the message was unsolicited

**07–0A** Message ID or 'USS'. If USS, the message ID appears in the variable text field in the MSG2 entry. If message ID, the 4-digit message number (or 3-digit message number and 1-character type code) appears in this field.

**0B–0F** Issuing module ID (4th, 5th, 6th, 7th, and 8th characters of the issuing module name). See "Module names in internal trace records" on page 21 for information on determining the module name from the module ID.

**10–13** Save area address

**14–17** Address of the CPMSG macro
- If the high order bit in the return address field is on, CPMSG has been issued from a utility module. The module name and return address are that of the caller of the utility module.
- If the high order bit in the return address field is off, CPMSG has not been issued from a utility module. The module name and return address are that of the module that issued the CPMSG.

**18–1F** Destination
- If byte 06 is **S**, this field contains one of the following:

  **SYSTEMxx**
  > *xx* is the ID of the system console.

  **NLDM**
  > Message is destined for the session monitor component of the NetView program or NLDM.

  **URC** Message is to be sent to the system console using routing codes, instead of to one console in particular.

  **POA** Message is to be sent to a program operator application (POA) and the POA name is not available yet.

- If byte 06 is **U**, this field is 0

## MSGS entry for message sent (Part 1)

**Entry:  MSGS**
**VIT option:**
> MSG

**Event:**  Message sent (Part 1)
**VIT processing module:**
> ISTRACOT

**Control is returned to:**
> ISTLUCRC or ISTMSCCN

This trace record is written each time VTAM sends a message to its destination. It can be used to correlate the console log with the internal trace. The destination may indicate that the message was sent to a system console (by way of a WTO macro) or to a program operator application, such as the NetView program (by way of a RCVCMD macro).

| 0 0 | 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B C D E F |
| MSGS | I D | F L A G S | 0 | DESTINATION | MESSAGE HEADER | FIRST 12 CHARACTERS OF MESSAGE TEXT |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"MSGS"
**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**      Message suppression indicator
      **0**      Message not suppressed.
      **1**      Message was suppressed.
**06–07**  0
**08–0F**  Destination. If **SYSTEMxx**, the message was sent to system console **xx**, by a WTO macro. If an application name, the message was sent to a program operator application, by a RCVCMD macro.
**10–13**  Message header (in hex). If destination is **SYSTEMxx**, this will be the WPL header. If destination is a program operator application, this will be a program operator header (ISTDPOHD).
**14–1F**  First 12 characters of the message text. If the message text is longer than 12 characters, it will continue in the MSG2 entry.

# MSG2 entry for message sent (Part 2)

**Entry:  MSG2**
**VIT option:**
    MSG
**Event:**  Message issued or message sent (Part 2)
**VIT processing module:**
    ISTRACOT

This trace record is a continuation of the MSG or MSGS trace record. It contains variable data for the MSG entry, or more message text for the MSGS entry.

Up to two MSG2 entries can follow the MSG entry; however, only one MSG2 entry can follow the MSGS entry.

| 0 0 | 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F 0 1 2 3 | 4 5 6 7 8 9 A B C D E F |
| MSG2 | | VARIABLE DATA OR MORE MESSAGE TEXT | |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"MSG2"
**04–1F**  Variable data, or more message text

## MT entry for module trace

**Entry: MT**
**VIT option:**
    SSCP
**Event:** Module trace detects full data buffer
**VIT processing module:**
    ISTRACOT
**Control is returned to:**
    ISTITCTR

This trace record is written when the module trace detects a full data buffer.

| 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 | 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| MT | DATA |

**Byte (hex)**
    **Contents**
**00–01**  Record ID: C"MT"
**02–1F**  Data

## MU1 entry for LU 6.2 message unit (Part 1)

**Entry: MU1**
**VIT option:**
    APPC
**Event:** LU 6.2 message unit (Part 1)
**VIT processing module:**
    ISTRACAC
**Control is returned to:**
    Module invoking the INTRACE macro that caused the record to be
    produced

This trace record shows information about an LU 6.2 message unit. It is generated whenever a message unit is queued to another component.

| 0 0 0 0 | 04 | 0 0 0 | 0 0 0 | 0 0 0 0 | 1 1 1 1 | 14 | 1 1 | 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | | 5 6 7 | 8 9 A | B C D E F | 0 1 2 3 | | 5 6 | 7 | 8 9 A B | C D E F |
| MU1 | I D | MU ID | MU ADDRESS | HALF SESS ID | CORR VALUE | R T N C D | 0 | F L A G | AMU FLAGS | SENSE DATA OR 0 |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"MU1"

**MU1**

| | |
|---|---|
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05–07** | Message unit ID |

X'010101'  (END_CONVERSATION)
X'010102'  (SESSION_FLOW_RESUMED)
X'010103'  (FLUSH_DATA)
X'010104'  (RESUME_PS_HS_FLOW)
X'010201'  (PS_COPR_FMH5_RCVD)
X'010301'  (SEND_EXPEDITED_DATA)
X'010302'  (CONFIRMED)
X'010303'  (REQUEST_TO_SEND)
X'010304'  (SEND_DATA_RECORD)
X'010305'  (SEND_ERROR)
X'010306'  (SEND_PACING_RSP)
X'010307'  (RSP_TO_EXPEDITED_DATA)
X'010501'  (ALLOCATE_RCB)
X'010502'  (DEALLOCATE_RCB)
X'010503'  (GET_SESSION)
X'010504'  (UNBIND_PROTOCOL_ERROR)
X'010505'  (REJECT_SESSION)
X'010506'  (SUSPEND_SESSION)
X'010507'  (RESUME_SESSION)
X'010701'  (ATTACH_TP)
X'010702'  (END_XP)
X'020101'  (CNOS_ABORT)
X'020201'  (CNOS_CLEANUP)
X'020501'  (CNOS_COMPLETE)
X'020502'  (CHANGE_SESSIONS)
X'030101'  (RECEIVE_EXPEDITED_DATA)
X'030102'  (CONFIRMED)
X'030103'  (REQUEST_TO_SEND)
X'030104'  (RECEIVE_DATA)
X'030105'  (RECEIVE_ERROR)
X'030106'  (PACING_RSP_RCVD)
X'030107'  (RSP_TO_REQUEST_TO_SEND)
X'030108'  (INITIAL_PACING_COUNT)
X'030109'  (DEALLOCATE_ABEND_REJECTED)
X'030401'  (ABORT_HS)
X'030501'  (ATTACH_HEADER)
X'030502'  (FREE_SESSION)
X'030503'  (BID)
X'030504'  (BID_RSP)
X'030505'  (BIS_RQ)
X'030506'  (BIS_REPLY)
X'030507'  (RTR_RQ)
X'030508'  (RTR_RSP)
X'030509'  (SECURITY_HEADER)
X'040101'  (RESTORE_SESSION)
X'040401'  (LOAD_PROFILES)
X'040402'  (RSP_LOAD_PROFILES)
X'040403'  (DELETE_PROFILES)
X'040501'  (SESSION_ACTIVATED)
X'040502'  (SESSION_DEACTIVATED)
X'040503'  (ACTIVATE_SESSION_RSP)
X'040504'  (CTERM_DEACTIVATE_SESSION)

|  |  |
|---|---|
| | X'040505' (LNS_LRM_FREE_AMU) |
| | X'050101' (RCB_ALLOCATED) |
| | X'050103' (SESSION_ALLOCATED) |
| | X'050104' (ATTACH_RECEIVED) |
| | X'050105' (CONVERSATION_FAILURE) |
| | X'050106' (SESSION_REJECTED) |
| | X'050107' (SESSION_SUSPENDED) |
| | X'050108' (SESSION_RESUMED) |
| | X'050109' (RESUME_SESSION_FLOW) |
| | X'050301' (HS_PS_CONNECTED) |
| | X'050302' (YIELD_SESSION) |
| | X'050303' (BID_WITHOUT_ATTACH) |
| | X'050304' (BID_RSP) |
| | X'050305' (BIS_RQ) |
| | X'050306' (BIS_REPLY) |
| | X'050307' (RTR_RQ) |
| | X'050308' (RTR_RSP) |
| | X'050309' (ENCIPHERED_RD2) |
| | X'050401' (ACTIVATE_SESSION) |
| | X'050402' (DEACTIVATE_SESSION) |
| | X'060201' (MODIFY_CNOS) |
| | X'060202' (MODIFY_DEFINE) |
| | X'060203' (DISPLAY_CNOS) |
| | X'060204' (DISPLAY_LUS) |
| | X'060205' (DISPLAY_MODES) |
| | X'060206' (DISPLAY_CONVS) |
| | X'060401' (MODIFY_PROFILES) |
| | X'070101' (TERMINATE_TP) |
| | X'070301' (SEND_1WAY_FASTPATH) |
| | X'080401' (REQ_LOAD_PROFILES) |
| **08–0B** | Message unit address |
| **0C–0F** | Half session ID |
| **10–13** | Address of correlator value |
| **14** | LU 6.2 message unit return code (AMURETCD) |
| **15–16** | 0 |
| **17** | Flag byte to indicate that the message unit contains an extended buffer list (bit 7) |
| **18–1B** | LU 6.2 message unit flags (AMUFLAGS) |
| **1C–1F** | LU 6.2 message unit sense data (AMUSENSE) or 0 |

# MU2 entry for LU 6.2 message unit (Part 2)

**Entry: MU2**
**VIT option:**
    APPC
**Event:** LU 6.2 message unit (Part 2)
**VIT processing module:**
    ISTRACAC

This trace record is a continuation of the MU1 trace record.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F |
| MU2 | 0 | MU ID | DATA ADDRESS OR 0 | DATA LENGTH OR 0 | RAB ADDRESS OR 0 | SES CNT OR 0 | SES LIM OR 0 | CONV ID OR 0 | VARIABLE OVERLAY FIELD |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"MU2"

**04** 0

**05–07** Message unit ID shown in MU1 entry

**08–0B** Data address or 0. If the message unit contains an extended buffer list, this field points to the address of the first extended buffer list entry that contains RU data.

**0C–0F** AMUDATLN (data length) or 0. If the message unit contains an extended buffer list, this field contains the length of all RU data represented in that extended buffer list.

**10–13** Resource allocation block address or 0

**14–15** For MU ID X'020502': Session count. For all other MU IDs: 0.

**16–17** For MU ID X'020502': Session limit. For all other MU IDs: 0.

**18–1B** For MU IDs associated with an active conversation (X'010102', X'010201', X'010506', X'030108', X'050107', X'050109', X'050301'): Conversation ID. For all other MU IDs: 0.

**1C–1F** Variable overlay field

For MU ID X'020201': COPR control block address

For MU ID X'020502':

**1C–1D**

Change in the CNOS value

**1E–1F** 0

For MU IDs X'030106' and X'030108': Pacing count increment

For MU IDs X'030501', X'030503', X'030504', and X'030507': RPH address for inbound half session

For MU IDs X'040402' and X'040403': ACEE address

For MU IDs X'010701': Address of the associated data for the MU

For all other MU IDs: 0

# MU3 entry for LU 6.2 message unit (Part 3)

**Entry: MU3**

**VIT option:**

APPC

**Event:** LU 6.2 message unit (Part 3)

**VIT processing module:**

ISTRACAC

This trace record is a continuation of the MU2 trace record. It is generated for only those message units which contain the following MU IDs:

- X'010102'
- X'010201'
- X'010501'
- X'010505'
- X'010506'
- X'020502'
- X'030509'
- X'040401'
- X'040402'
- X'040403'
- X'040501'
- X'050106'
- X'050107'
- X'050108'
- X'050109'
- X'050309'
- X'050401'
- X'060201'
- X'060202'
- X'060203'
- X'060204'
- X'060205'
- X'060206'
- X'060401'
- X'080401'

| 0 0 0 0<br>0 1 2 3 | 0<br>4 | 0 0 0<br>5 6 7 | 0 0 0 0 0 0 0 0<br>8 9 A B C D E F | 1 1 1 1 1 1 1 1<br>0 1 2 3 4 5 6 7 | 1 1 1 1 1 1 1 1<br>8 9 A B C D E F |
|---|---|---|---|---|---|
| MU3 | 0 | MU ID | MODE NAME OR<br>SESSION ID OR 0 | LOCAL LU NAME | PARTNER LU NAME |

**Byte (hex)**
    **Contents**
**00–03**    Record ID: C"MU3"
**04**    0
**05–07**    Message unit ID shown in the MU1 and MU2 entries
**08–0F**    Mode name for the following MU IDs:
        X'020502'
        X'040501'
        X'050401'
        X'060202'
        X'060203'

        Mode name or 0 for the following MU IDs:
        X'010501'
        X'060201'

X'060205'
X'060206'

Session instance identifier for the following MU IDs:
X'010102'
X'010506'
X'010507'
X'050106'
X'050107'
X'050108'
X'050109'

Session instance identifier or 0 for the following MU ID:
X'010505'

0 for all other MU IDs

**10–17**   Local logical unit name
**18–1F**   Partner logical unit name or 0; for X'060204', 0

# MU4 entry for LU 6.2 message unit (Part 4)

**Entry:   MU4**
**VIT option:**
        APPC
**Event:   LU 6.2 message unit (Part 4)**
**VIT processing module:**
        ISTRACAC

This trace record is a continuation of the MU3 trace record. It is generated for only
those message units that contain the following MU IDs:

- X'010102'
- X'010201'
- X'010501'
- X'010505'
- X'010506'
- X'020502'
- X'030509'
- X'040401'
- X'040402'
- X'040403'
- X'040501'
- X'050106'
- X'050107'
- X'050108'
- X'050109'
- X'050309'
- X'050401'
- X'060201'
- X'060202'
- X'060203'
- X'060204'

- X'060205'
- X'060206'
- X'060401'
- X'080401'

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| MU4 | 0 | PARTNER LU NETID OR 0 |

**Byte (hex)**
    **Contents**
**00–03**    Record ID: C"MU4"
**04–17**    0
**18–1F**    Partner network identifier or 0; for X'060204', 0.

# NIPx entry for IP characteristics list add, delete, or find

**Entry:**  **NIPA, NIPD, NIPI, or NIPN**
**VIT option:**
    NRM
**Event:**  Call to module ISTNRCIP
**VIT processing module:**
    ISTRACNR
**Control is returned to:**
    ISTNRCIP

This trace record contains information about IP characteristics being added, deleted, or found.

| 0 0 0 0 | 0 0 | 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| NIPA NIPD NIPI NIPN | I D | 0 F L A G S R C | LU NAME | CALLER ADDRESS | NETWORK ID | RPH ADDRESS |

**Byte (hex)**    **Contents**
**00–03**    Record ID:
        C"NIPA" (add)
        C"NIPD" (delete)
        C"NIPI" (find by IP address)
        C"NIPN" (find by name)
**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**    0
**06**    Flags

        **Bit**            **Meaning**

<table>
<tr><td>1111 11..</td><td>0</td></tr>
<tr><td>.... ..1.</td><td>LU is an APPL</td></tr>
<tr><td>.... ...1</td><td>LU is TN3270 client supporting definite response mode</td></tr>
</table>

| | |
|---|---|
| 07 | Return code |
| 08–0F | LU name |
| 10–13 | Address of the caller of ISTNRCIP |
| 14–1B | Network ID of the LU |
| 1C–1F | RPH Address |

## NIP2 entry for IP characteristics list add, delete, or find (Part 2)

**Entry: NIP2**
**VIT option:**
　　NRM
**Event:** Call to module ISTNRCIP
**VIT processing module:**
　　ISTRACNR

This trace record is a continuation of the NIPx entry.

| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|

| NIP2 | IP ADDRESS | PORT | 0 | IPADR ADDRESS | RDTE ADDRESS |
|---|---|---|---|---|---|

| Byte (hex) | Contents |
|---|---|
| 00–03 | Record ID: C'NIP2' |
| 04–13 | IP address |
| 14–15 | Port number |
| 16–17 | 0 |
| 18–1B | Address of IP address structure (ISTIPADR) |
| 1C–1F | RDTE address |

---

# NLPx entry for network layer packet (Part 1)

**Entry: NLP**
**VIT option:**
　　HPR
**Event:** Network layer packet (NLP) sent or received
**VIT processing module:**
　　ISTITCHP
**Control is returned to:**
　　The module that issued the INTRACE macro

This trace record is written when a network layer packet (NLP) is sent or received at the DLC layer. The NLPI and NLPO entries provide, respectively, inbound and outbound information about the NLPs, including the FID5 PIU and information about the network layer header (control block ISTNLH) and the rapid transport protocol (RTP) transport header (control block ISTTHDR). The NLP record is 32 bytes in length, with up to 31 additional NLP2 continuation records, each 32 bytes

in length.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| NLPI OR NLPO | | I D | 0 | T S C I L N G | TSCB ADDR | | DLC NCB ADDR | | RPNCB ADDR | | RELATED TSCB ADDR | | ISSUING MODULE NAME | | RPH ADDR | |

**Byte (hex)**
**Contents**

**00–03**   Record ID:
C"NLPI" for inbound
C"NLPO" for outbound

**04**       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**       0

**06–07**   Length of data (TSCILNG)

**08–0B**   TSCB address

**0C–0F**   DLC NCB address

**10-13**   RPNCB address

**14–17**   Related TSCB address:
Continuation TSCB for inbound (0 if no continuation TSCB exists).
Data TSCB for outbound (first TSCB in chain with TSCNLP off, 0 if no data TSCB exists).

**18–1B**   Name of the module that issued this trace entry

**1C–1F**   RPH address

# NLP2 entry for network layer packet (Part 2)

**Entry:  NLP2**
**VIT option:**
HPR
**Event:**  Network layer packet (NLP) sent or received
**VIT processing module:**
ISTITCHP

This trace record is a continuation of the NLP entry.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| NLP2 | | 28 BYTES OF NLP | | | | | | | | | | | | | |

**Byte (hex)**
**Contents**
**00–03**   Record ID: C"NLP2"

04–1F  28 bytes of the network layer packet (NLP), including the network layer header (NHDR), the RTP transport header (THDR), and the PIU

## NRSP entry for negative response to PIU request (Part 1)

**Entry:  NRSP**
**VIT option:**
PIU
**Event:**  Negative response to PIU request (Part 1)
**VIT processing module:**
ISTRACOT
**Control is returned to:**
ISTTSCGR

This trace record is written when VTAM generates a negative response to a PIU request. Because this entry is associated with an event failure, it is treated as an exception condition and is always traced, regardless of the VIT options specified, if the VIT is active.

| 0 0 | 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| NRSP | | I D | C B I D | F L A G S | 0 | TSCB ADDRESS | 20 BYTES OF PIU |

**Byte (hex)**
**Contents**
**00–03**  Record ID: C"NRSP"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  Control block ID
**06**  Flag field (TSCFLAG1)
**07**  0
**08–0B**  Address of TSCB
**0C–1F**  First 20 bytes of the PIU

## NRS2 entry for negative response to PIU request (Part 2)

**Entry:  NRS2**
**VIT option:**
PIU
**Event:**  Negative response to PIU request (Part 2)
**VIT processing module:**
ISTRACOT

This trace record is a continuation of the NRSP entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| NRS2 | 24 MORE CHARACTERS OF THE PIU | RETURN ADDRESS |

**Byte (hex)**

**Contents**

00–03  Record ID: C"NRS2"

04–1B  24 more characters of the PIU

1C–1F  Return address of the issuer

# NSD entry for NSIND exit

**Entry: NSD**

**VIT option:**
  VCNS

**Event: NSIND exit scheduled**

**VIT processing module:**
  ISTRACNS

**Control is returned to:**
  ISTAPCUE

This trace record is written when the NSIND exit routine is scheduled. The exit is scheduled when VTAM receives a VCNSCMD CONTROL=LOGON request from a VCNS user. This trace record shows information about the exit invocation.

| 0 0 0 0 | 0 4 | 0 5 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| NSD | I D | 0 | ACB ADDR | NETWORK ACCESS POINT | EXIT REASON | STORAGE ADDRESS | ERROR INFO |

**Byte (hex)**

**Contents**

00–03  Record ID: C"NSD"

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05–07  0

08–0B  Address of the ACB associated with the NSIND exit

0C–13  Symbolic name of the network access point, defined with USER=VCNS in the interconnect major node, representing a line to which the application program is logged on

14–17  Reason the exit was scheduled:

    **0**      As a result of a network management-related event

    **4**      As a result of a termination of the VCNS line

    **8**      As a result of an error detected by VTAM that resulted in the termination of the LOGON request (for example, an abend)

18–1B  If exit reason in bytes 14–17 is 0, address of storage area containing specific error information about the exit event; otherwise, 0.

**1C–1F** If bytes 18–1B contain a storage area address, first 4 bytes of specific error information from the storage area.

---

# ODPK entry for OSA-Express QDIO or HiperSockets packets (Part 1)

**Entry: ODPK**
**VIT option:**
> CIA

**Event:** Inbound or outbound data
**VIT processing module:**
> ISTITCOD

**Control is returned to:**
> ISTLLCIE, ITSLLCWI

This trace record is written when packets are read from or written to an OSA-Express QDIO or HiperSockets adapter.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 | 1 | 1 | 1 1 | 1 1 | 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 | 5 | 6 | 7 8 | 9 A | B | C D E F |
| ODPK | I D | 0 | D E V I C E T Y P E | D I R E C T I O N | SPAC ADDRESS | NCB ADDRESS | XBFL ADDRESS | P R I O R I T Y | S B A L | S T A R T I N D E X | E N D I N D E X | P A C K E T | L E N G T H | M O D U L E | I D E N T I F I E R | RPH ADDRESS |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"ODPK"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Device type:
- If byte 7 is C"R":
  - C"I" for HiperSockets (iQDIO)
  - C"Q" for QDIO
- If byte 7 is C"I" or C"O", byte 06 is 0

**07** Direction:
- C"O" for outbound
- C"I" for inbound
- C"R" for HiperSockets Accelerator, outbound

**08–0B** ShortPAC address
**0C–0F** DINCB address
**10–13** XBUFLST entry address
**14** Queue priority
**15** SBAL index
**16** Start SBALE index (within SBAL)
**17** End SBALE index (within SBAL)
**18–19** Total length of packet
**1A–1B** Module identifier for the module that issued the INTRACE.
**1C–1F** Request parameter header (RPH) address.

## ODP2 entry for OSA-Express QDIO or HiperSockets packets (Part 2)

**Entry: ODP2**
**VIT option:**
> CIA

**Event:** Inbound or outbound data
**VIT processing module:**
> ISTITCOD

This trace record is written following ODPK and may be written following another ODP2 depending on the amount of data to be traced.

**Tip:** The number of ODP2 records captured for each packet is limited. When running an IO trace, the number of ODP2 records will vary with the length specified on the MODIFY TRACE command. For a data path channel used for capturing OSA-Express network traffic analyzer traces, only one ODP2 record will be captured.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| ODP2 | 28 BYTES OF PACKET DATA | | | | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"ODP2"
**04–1F** If preceded by ODPK, first 28 bytes of packet.

> If preceded by ODP2, next 28 bytes of packet.

# ODTE entry for OSA-Express QDIO or HiperSockets timer events

**Entry: ODTE**
**VIT option:**
> CIA

**Event:** OSA-Express QDIO or HiperSockets timer
**VIT processing module:**
> ISTITCOD

**Control is returned to**
> ISTLLCWD, ISTLLCWT

This trace record is written when the OSA-Express QDIO or HiperSockets write completion timer is started or restarted.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| ODTE | | | | ID | | 0 | | F U N C T I O N | | | CURRENT TIME OF DAY | | | | | | INTERVAL | | | | | | R E T U R N | C O D E | M O D U L E | I D E N T I F I E R | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"ODTE"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–06** 0

**07** Function:
- C"S" for start timer
- C"R" for restart timer

**08–0F** Current Timer of Day (in TOD format)

**10–17** Interval (in TOD format)

**18–19** STIMERM return code

**Code value**

**Meaning**

**X'0000'**

STIMERM service completed successfully.

**X'000C'**

Program error: Interval exceeds 2400 hours.

**X'0010'**

Program error: Parameters that are not valid are passed to STIMERM.

**X'001C'**

Program error: STIMERM SET limit for task exceeded.

**X'0024'**

Program error: Not valid STIMERM ID.

**X'0028'**

Program error: Interval plus TOD exceeds maximum clock comparator value.

**1A–1B** Module identifier for the module that issued the INTRACE.

**1C–1F** Request parameter header (RPH) address.

# ONLP entry for orphaned Network Layer Packet

**Entry: ONLP**

**VIT option:**

HPR

**Event: Orphaned NLP**

**VIT processing module:**
>ISTITCHP

**Control is returned to:**
>Module issuing INTRACE

An orphaned NLP is an NLP that appears to have been lost by the DLC.

This trace record is written when an orphaned NLP is detected, recovered, or freed. There are two types of orphaned NLPs, acknowledged and unacknowledged.

| 0 0 | 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 | 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| O N L P | A S I D | 0 | F u n c | R c v C n t | NLP Seqnum | SND Address | Caller Address | TSCB Address | RPNCB Address | SGMNT Address |

**Byte (hex)**

>**Contents**

| | |
|---|---|
| 00–03 | Record ID: C"ONLP" |
| 04 | ID is the primary address spaced ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | ONLP function: |

>**A**     Acknowledged orphaned NLP detected

>**F**     Free acknowledged orphaned NLP from Garbage_SNDs queue

>**R**     Recover unacknowledged orphaned NLP

>**S**     Segment recovered NLP

>**U**     Unacknowledged orphaned NLP detected

| | |
|---|---|
| 07 | Recovery count |
| 08–0B | NLP sequence number |
| 0C-0F | ISTSND address |
| 10–13 | Address of INTRACE issuer |
| 14–17 | ISTTSCB address |
| 18–1B | ISTRPNCB address |
| 1C-1F | ISTSGMNT address or zeros |

# OON entry for module trace

**Entry: OON**

**VIT option:**
>N/A

**Event:** Method has been entered.

**VIT processing module:**
>ISTRACOT

**Control is returned to:**
>Module invoking the INTRACE macro that caused the record to be produced

This trace record is written when a method is entered.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| OON | | | 0 | MODULE NAME | | | | | METHOD NAME | | | | | | | | | | | | | | | | | | | RPH or Reg1 | | | |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"OON"

**03** 0

**04–08** Name of the module containing the invoked method

**09–1B** Method name

**1C-1F** RPH address or register 1

# OON2 entry for module trace

**Entry: OON2**

**VIT option:**

N/A

**Event:** Method has been entered

**VIT processing module:**

ISTRACOT

**Control is returned to:**

Module invoking the INTRACE macro that caused the record to be produced

This trace record is a continuation record of the OON trace record. Multiple OON2 records will be present if the OO method being traced has more than six input parameters.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| OON2 | | | | PARM ADDR or 0 | | | | PARM ADDR or 0 | | | | PARM ADDR or 0 | | | | PARM ADDR or 0 | | | | PARM ADDR or 0 | | | | PARM ADDR or 0 | | | | RETN ADDR or PARM ADDR or 0 | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"OON2"

**04–1B** Addresses of the parameters passed to the OO method or 0

**1C-1F** Return or invocation address of the requester of the OO method or address of the parameter passed to the OO method or 0

**Note:** One word for each parameter passed to the OO method will be listed in the trace entry. If the parameter is not present, it will have the value of 0. The first OON2 trace entry has the structure address at offset 4 and has the return or the invocation address of the OO method at offset X'1C'.

Additional OON2 trace entries will have a parameter address at offset X'1C', instead of the return or invocation address.

# OOSQ entry for HPR out-of-sequence queue sequence numbers

**Entry:  OOSQ**
**VIT option:**
> HPR

**Event:**  Out-of-sequence queue sequence numbers
**VIT processing module:**
> ISTITCHP

This trace record shows the HPR sequence numbers in all the DAPTRs on the HPR out-of-sequence queue. Each sequence number pair represents the sequence numbers that have been received and queued. The first gap is between the LAST BYTE RCVD (refer to "RTP entry for RTP PAB dispatch" on page 320) and the begin sequence number of the first sequence number pair. Additional gaps exist between the end sequence number of one pair and the begin sequence number of the next pair. If there are more than 2 gaps, this record is followed by as many OOS2 records as required to show all the sequence numbers on the RPNCB out-of-sequence queue.

The OOSQ record is written at the start of an RTP PAB dispatch when at least one DAPTR is on the out-of-sequence queue. OOSQ is also written at the end of an RTP PAB dispatch when at least one of the DAPTRs on the out-of-sequence queue has been altered during the dispatch.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F | 10 11 12 13 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|
| OOSQ | ASID | 0 | FLAG1 | FLAG2 | SEQUENCE NUMBER PAIR 1 | SEQUENCE NUMBER PAIR 2 (OR ZEROS) | RPNCB ADDRESS | RPH ADDRESS |

**Byte (hex)**
> **Contents**

**00–03**  Record ID: C"OOSQ"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  0
**06**  DAP_Flags for sequence number pair 1
**07**  DAP_Flags for sequence number pair 2 (zeros if only a single pair exists)
**08–0F**  First sequence number pair

- Word 1  Begin Sequence number
- Word 2  End Sequence number

**10–17**  Second sequence number pair (zeros if only a single pair exists)

- Word 1  Begin Sequence number
- Word 2  End Sequence number

**18–1B**  RPNCB address
**1C–1F**  Request parameter header (RPH) address

## OOS2 entry for HPR out-of-sequence queue sequence numbers (Part 2)

**Entry: OOS2**
**VIT option:**
> HPR

**Event:** Out-of-sequence queue sequence numbers (Part 2)
**VIT processing module:**
> ISTITCHP

Any number (including 0) of OOS2's may follow the OOSQ, depending on the number of DAPTRs on the HPR out-of-sequence queue.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F | 10 11 12 13 14 15 16 17 | 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|
| OOS2 | RECCNT | FLAGN | FLAGN+1 | FLAGN+2 | SEQUENCE NUMBER PAIR N | SEQUENCE NUMBER PAIR N+1 (OR ZEROS) | SEQUENCE NUMBER PAIR N+2 (OR ZEROS) |

**Byte (hex)**
> **Contents**

| | |
|---|---|
| 00–03 | Record ID: C'OOS2' |
| 04 | OOS2 record counter |
| 05 | DAP_Flags for sequence number pair n |
| 06 | DAP_Flags for sequence number pair n +1 (zeros if pair does not exist) |
| 07 | DAP_Flags for sequence number pair n +2 (zeros if pair does not exist) |
| 08–0F | Next sequence number pair |

- Word 1 Begin sequence number
- Word 2 End sequence numbers

| | |
|---|---|
| 10–17 | Next sequence number pair (zeros if pair does not exist) |

- Word 1 Begin sequence number
- Word 2 End sequence number

| | |
|---|---|
| 18–1F | Next sequence number pair (zeros if pair does not exist) |

- Word 1 Begin sequence number
- Word 2 End sequence number

# OOX entry for module trace

**Entry: OOX**
**VIT option:**
> N/A

**Event:** Method has been exited.
**VIT processing module:**
> ISTRACOT

**Control is returned to:**
> Module invoking the INTRACE macro that caused the record to be produced

This trace record is written when a method is exited.

| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|
| 0 1 2 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |

| OOX | R C | MODULE NAME | METHOD NAME | RPH or Reg1 |
|---|---|---|---|---|

**Byte (hex)**
**Contents**
**00–02** Record ID: C"OOX"
**03** Return code
**04–08** Name of the module containing the invoked method
**09–1B** Method name
**1C-1F** RPH address or register 1

# OPER entry for operator command (Part 1)

**Entry: OPER**
**VIT option:**
    MSG
**Event:** Operator command (Part 1)
**VIT processing module:**
    ISTRACOT
**Control is returned to:**
    ISTLUCMD, ISTCFF3D, ISTINCF9

This trace record is written each time a VTAM command is issued at the operator console. This can help you match the console log to a surge of activity shown in the VIT. This trace record is also useful when the console log is unavailable. Additionally, it provides a record of POA commands that have been issued. These do not appear on the console log.

The variable data in bytes 08–1F is the command text, with minor modifications. If this data is longer than 24 characters, it will continue in the OPE2 entry. The command always appears in abbreviated form:
**D** DISPLAY command
**F** MODIFY command
**V** VARY command
**Z** HALT command
**?** Unknown command passed to VTAM

NET and procname are excluded from the command text. Because the VTAM internal trace is not active when VTAM is started, the START command is not traced.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPER | | | | ID | FLAG | 0 | | FIRST 24 CHARACTERS OF OPERATOR COMMAND | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"OPER"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Flag byte

 **Bit**  **Meaning**
 **1... ....**
   A POA command was issued.
 **0... ....**
   Not a POA command.
 **.1.. ....**
   A POA needs the command complete message.
 **.0.. ....**
   No command complete message for POA.
**06–07** 0
**08–1F** First 24 characters of the operator command

## OPE2 entry for operator command (Part 2)

**Entry: OPE2**
**VIT option:**
 MSG
**Event:** Operator command (Part 2)
**VIT processing module:**
 ISTRACOT

This trace record is a continuation of the OPER entry.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OPE2 | | | | UP TO 28 MORE CHARACTERS OF THE OPERATOR COMMAND | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"OPE2"
**04–1F** Up to 28 more characters of the operator command

## ORMG entry for storage management

 **Entry: ORMG**

**VIT option:**
> SMS

**Event:** ISTORMMG Processing

**VIT processing module:**
> ISTRACSM

**Control is returned to:**
> ISTORMMG

This trace record records various storage management values at critical points in ISTORMMG.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| ORMG | ID | 0 | INST | 0 | BPD-OBFOR | BPD-OBACK | ATC-OROBT | ATC-ORTBF | ATC-CSAFR | ATC-CSAFI |

**Byte (hex)**

> **Contents**

| | |
|---|---|
| 00–03 | Record ID: C"ORMG" |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | The instance of the trace record in the module |
| 07 | 0 |
| 08–0B | BPDOBFOR–Double-threaded obtained storage queue forward pointer |
| 0C–0F | BPD0BACK–Double-threaded obtained storage queue backward pointer |
| 10–13 | ATCOROBT–Single-threaded obtained storage queue pointer |
| 14–17 | ATCORTBF–To be freed queue pointer |
| 18–1B | Free CSA bytes on the obtained queue |
| 1C–1F | ATCCSAFI–Free CSA requests on the obtained queue |

# PAGB entry for PAGE_BUFFER requests

**Entry: PAGB**

**VIT option:**
> CSM

**Event:** IVTCSM REQUEST=PAGE_BUFFER

**VIT processing module:**
> ISTITCCS

**Control is returned to:**
> IVTSMCBF

This trace record provides the status of an IVTCSM REQUEST=PAGE_BUFFER macroinstruction.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAGB | | | | I D | 0 | F L A G | R E C N U M | T Y P E | 0 | | | RTN COD | | RSN COD | | UTILRTN CALLER OR RETURN ADDRESS | | | | LAST BUFFER LIST ENTRY | | | | NUMBER OF BUFFERS | | | | THREAD VALUE OR 0 | | | |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"PAGB"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace record flag:

**B'0... ....'**
Indicates that this is the last record for this event.

**B'1... ....'**
Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07** Trace record number to correlate all the entries for this particular event.

**08** Type or state of buffers allocated
**X'80'** Fixed
**X'20'** Eligible to be made pageable

**09–0B** 0

**0C–0D**
Return code

**0E–0F** Reason code

**10–13** Address of utility routine caller or return address of the issuer of the IVTCSM macroinstruction

**14–17** Address of the buffer list entry that was being processed when the error was encountered.

**18–1B** Number of buffers

**1C–1F** THREAD value if specified or 0 if THREAD is not specified. The THREAD value is used only to correlate this trace record to a specific IVTCSM macroinstruction.

## PAG2 entry for PAGE_BUFFER requests

**Entry: PAG2**
**VIT option:**
CSM
**Event: IVTCSM REQUEST=PAGE_BUFFER**
**VIT processing module:**
ISTITCCS

This trace record is a continuation of the PAGB trace record. Each PAG2 record traces, at most, two buffers that were requested to be placed in a pageable state.

| 0 0 | 0 0 | 0 0 | 0 | 0 | 0 0 0 0 0 0 0 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 | 7 | 8 9 A B C D E F 0 1 2 3 | 4 5 6 7 8 9 A B C D E F |
| PAG2 | | 0 | F L A G | R E C N U M | BUFFER TOKEN FOR INPUT BUFFER LIST ENTRY | BUFFER TOKEN FOR INPUT BUFFER LIST ENTRY OR 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"PAG2"

**04–05** 0

**06** Trace record flag:

**B'0... ....'**
Indicates that this is the last record for this event.

**B'1... ....'**
Indicates that additional records exist for this event. Use the trace record number from this entry to locate corresponding continuation records.

**07** Trace record number to correlate all the entries for this particular event.

**08–13** Buffer token contained in input buffer list entry

**14–1F** Buffer token contained in input buffer list entry or 0

# PCIx entry for program-controlled or suspend interrupt

**Entry: PCID, PCIT, or PCIX**

**VIT option:**
CIO

**Event:** Program-controlled or suspend interrupt

**VIT processing module:**
ISTITCOD

**Control is returned to:**
ISTTSCIE

This trace record is written when a program-controlled interrupt occurs.

This interrupt occurs for a CLAW channel-to-channel attached host, for HPDT read and write devices, or for the OSA-Express QDIO or HiperSockets adapter read queue.

The PCID entry is recorded when the OSA-Express QDIO or HiperSockets adapter has completed a read operation. The PCID entry may or may not be preceded by a SIGA (read) operation for the same device.

The PCIT and PCIX are correlated to the SIOx, RIOx, and INTx entries for the same device using the CUA field. The combination of the information provided by these entries describe the channel program management and I/O operations for the device.

Refer to *z/OS Communications Server: SNA Data Areas Volume 1* for a description of the NCB fields.

These events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

## PCIX and PCIT mapping and field descriptions

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCIX PCIT | | | | ID | 0 | STATE | OPCD | CUA DEVICE | | | | NCB ADDRESS | | | | FLAGS | | | | IOS | CODE | SECP | | CSW | | | | | | | |

**Byte (hex)**

**Contents**

00–03 Record ID: C"PCI"
04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05 Reserved
06 Link station state (NCBLNKST)
07 Operation code
08–0B Channel device name in EBCDIC (either a device address or device number)
0C–0F NCB address
10–13 Flag bytes (NCBFLAGS)
14 IOS Bit 1 IOSSSPND
   **X'80'** IOS suspended
15 I/O completion code (IOSCOD)
16–17 CPNCB_Index_Word
18–1F Channel status word from IOSB.

## PCID mapping and field descriptions

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PCID | | | | ID | 0 | | Func | CUA Address | | | | NCB Address | | | | Next | Last | Count | Stat | Count | Stat | Count | Stat | Count | Stat | Count | Stat | Count | Stat | Count | Stat |

**Byte (hex)**

**Contents**

00–03 Record ID: C"PCID" for DINCB
04 ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05–06 0
07 Function:
   • C"R" for real PCI interrupt
   • C"V" for virtual PCI interrupt

| 08–0B | Device address in EBCDIC (as specified in the TRL deck) |
|---|---|
| 0C–0F | DINCB address |
| 10 | Content of DINCB_Q_Data_Index_Next (Next Empty Read) |
| 11 | Content of DINCB_Q_Data_Index_Last (Last Empty Read) |
| 12–13 | Start of Count/Status Pairs |
| | Starting at SLSB(1), these fields represent the number of consecutive SLSBs containing the same status, and what the status value is. See SLS_SLSB_Status constants for status values. |
| 14–15 | Continuation Count/Status pairs (if applicable) |
| 16–17 | Continuation Count/Status pairs (if applicable) |
| 18–19 | Continuation Count/Status pairs (if applicable) |
| 1A–1B | Continuation Count/Status pairs (if applicable) |
| 1C–1D | |
| | Continuation Count/Status pairs (if applicable) |
| 1E–1F | Continuation Count/Status pairs (if applicable) |

## PIU entry (Part 1)

**Entry:** PIU
**VIT option:**
> PIU

**Event:** PIU record (Part 1)
**VIT processing module:**
> ISTRACOT

**Control is returned to:**
> ISTLSC6V ISTTSCPR, ISTTSCLS, ISTTSCRI, ISTTSCSC, ISTTSCSR, ISTTSCWS, ISTTSCPD, ISTTSCLE, ISTTSCXS (or ISTINCF1 and INTINCS1 for RUPE)

This trace record provides information about external and internal FID4 PIUs.

For most PIUs, only the first 48 bytes are traced (a 26-byte TH, a 3-byte RH, and the first 19 bytes of RU). However, certain classes of PIUs, such as session control RUs and VTAM RUs, are traced in their entirety.

If the control block is a RUPE, the TH is reconstructed by the internal trace using what information is available. For TSCB entries, the PIU is recorded as is.

If the PIU is a response with sense data, this entry will be generated whether the PIU option is in effect or not. It is treated as an exception condition and is always traced if the VIT is active, regardless of the VIT options specified.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| PIU | | | | I D | C B I D | TSCB OR RUPE INFO | | TSCB OR RUPE ADDRESS | | | | 20 BYTES OF PIU | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"PIU"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID: X'54'=RUPE, X'99'=TSCB

**06-07** The control block identifier in byte 05 determines the meaning of this field.

TSCB or RUPE information:

- If byte 05 indicates a TSCB, use the following values:
  - Byte 06 contains flags (TSCFLAG1).
  - Byte 07 contains a unique instance identifier.
- If byte 05 indicates a RUPE, use the following value:
  - Bytes 06-07 contain the index for a network element address.

**08–0B** TSCB or RUPE address (see byte 05)

**0C–1F** 20 bytes of PIU

# PIU2 entry (Part 2)

**Entry: PIU2**

**VIT option:**

PIU

**Event:** PIU record (Part 2)

**VIT processing module:**

ISTRACOT

This trace record is a continuation of the PIU entry and contains 28 more bytes of FID4 PIU.

The VTAM internal trace will generate as many as 31 PIU2 records in the following situations:

- For PIUs that are formatted and contain session control RUs
- For RUs to or from the SSCP or PUs
- When the VIT LCS option was specified and the PIU contains LCS data

The number of records generated depends on the data count field in the transmission header (TH).

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| PIU2 | 28 MORE BYTES OF PIU ||||||| 

**Byte (hex)**
　　　　**Contents**
**00–03**　Record ID: C"PIU2"
**04–1F**　Up to 28 more bytes of the FID4 PIU (padded at right with zeros)

---

# PKI or PKO entry for TCP/IP packet sent or received

**Entry:  PKI or PKO**
**VIT option:**
　　　　CIA
**Event:**  TCP/IP packet sent or received
**VIT processing module:**
　　　　ISTRACCI
**Control is returned to:**
　　　　ISTTCCXD, ISTTCCXB, ISTTCCWD, ISTTCCWB, ISTTCCLB, ISTTCCLD,
　　　　ISTTCCCB, ISTTCCCD

This trace entry provides information about packets received by or sent from
TCP/IP data link control (DLC).

A PKI trace record is written whenever the buffer control channel program
completes with packets received.

A PKO trace record is written whenever a packet is transferred to the
telecommunications subsystem or communication adapter by inserting it into the
buffer control channel program.

| 0 0 0 0 | 0 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| PKI PKO | I D | R E S E R V E D | P A C K E T | L E N G T H | T I P A C | A D D R E S S | PACKET DATA |||

**Byte (hex)**
　　　　**Contents**
**00–03**　Record ID:
　　　　C"PKI" for inbound packet data
　　　　C"PKO" for outbound packet data
**04**　　ID is the primary address space ID (ASID). This field is 0 if the ID is
　　　　greater than X'FF'.
**05**　　RESERVED

06–07   Total length of the packet
08–0B   TIPAC address
0C–1F   First 20 bytes of packet data

## PKI2 or PKO2 entry for TCP/IP packet sent or received

**Entry:  PKI2 or PKO2**
**VIT option:**
      CIA
**Event:**  TCP/IP packet sent or received
**VIT processing module:**
      ISTRACCI

This trace entry is a continuation of PKI and PKO.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| PKI2 PKO2 | 28 BYTES MORE OF PACKET DATA |

**Byte (hex)**
      **Contents**
00–03   Record ID:
          C"PKI2" for inbound packet data
          C"PKO2" for outbound packet data
04–1E   Next 28 bytes of packet data

**Note:** There is a maximum of seven PKI2 or PKO2 records.

## PLOQ entry for PLO queue manager event (IUTPLOQM macro)

**Entry:  PLOQ**
**VIT option:**
      CIA
**Event:**  Queue management using IUTPLOQM
**VIT processing module:**
      ISTITCOD
**Control is returned to**
      IUTPLOQM issuer

This trace record is written when the IUTPLOQM macro is executed. It is primarily
used to identify the addition or removal of elements to or from a queue header.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| PLOQ | I D   0   Return Code   Function | Element address | Before Synch / After Synch / Element / Count | Return Address | New queue head pointer | New queue tail pointer | PLOQH Address |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"PLOQ"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** IUTPLOQM return code

Add_Chain return codes:

**Code Value**
>    **Meaning**

**X'00'** Add_Chain successful and Synch_Byte mask operation was performed (if specified).

**X'04'** Add_Chain successful but Synch_Byte mask operation was not performed (LE|GE failed).

Remove_Element return codes:

**Code Value**
>    **Meaning**

**X'00'** Remove_Element successful and Synch_Byte mask operation was performed (if specified.)

**X'04'** Remove_Element successful but Synch_Byte mask operation was not performed (LE|GE failed).

**X'08'** Remove_Element unsuccessful but Synch_Byte mask operation was performed (if specified).

**X'0C'** Remove_Element unsuccessful and Synch_Byte mask operation was not performed (LE|GE failed).

Synch_Byte return codes:

**Code Value**
>    **Meaning**

**X'00'** Synch_Byte mask operation was performed.

**X'04'** Synch_Byte mask operation was not performed (LE|GE failed).

**07** Function:
- C"A" for Add_Chain

- C"R" for Remove_Element
- C"S" for Synch_byte_only

**08–0B** Element address:
- For Add_Chain, address of first element in chain
- For Remove_Element, address of element removed
- For Synch_byte_only, 0

**0C** Contents of Synch byte before PLO
**0D** Contents of Synch byte after PLO
**0E–0F** Contents of count field after PLO
**10–13** Address of caller (within IUTPLOQM expansion)
**14–17** Address of first element on queue after the PLO
**18–1B** Address of last element on queue after the PLO
**1C–1F** PLO Queue manager Header address (IUTPLOQH)

# POOF entry for freeing storage from GETBLK pool

**Entry: POOF**
**VIT option:**
    SMS
**Event:** POOLFREE macro called to free all storage in a GETBLK pool
**VIT processing module:**
    ISTRACSM
**Control is returned to:**
    Module invoking the INTRACE macro that caused the record to be
    produced

This trace record is generated when the POOLFREE macro uses the VTFREE macro to free all storage that was allocated with GETBLK. This trace record helps the VIT analysis tool determine that storage is freed.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| POOF | I D | 0 | P O O L | R C | 0 | 0 | RETURN ADDRESS | LENGTH FREED | 0 | RPH ADDRESS |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"POOF"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Pool type. The value X'7A' indicates the CMIPPVT storage pool type. For other possible storage pool types, refer to "FBLK entry for FREEBLK macro (Part 1)" on page 191 or "GBLK entry for GETBLK macro (Part 1)" on page 198. For more information on storage pools, refer to *z/OS Communications Server: SNA Network Implementation Guide*.
**07** Return code from the POOLFREE macro
**08–0F** 0
**10–13** Address of the issuer of the POOLFREE macro
**14–17** Length of storage freed, including the GETBLK headers. (Because the

headers are not included in GBLK length fields, the length of storage freed in the POOF VIT entry will be larger than the amount of storage reported in the GBLK VIT entries.)

**18–1B**   0

**1C–1F**   Request parameter header (RPH) address

## POST entry for post waiting event

**Entry:  POST**

**VIT option:**
> PSS

**Event:**  Post waiting event

**VIT processing module:**
> ISTRACPS

**Control is returned to:**
> ISTAPCTP

This trace record identifies an RPH that is being posted (using the TPPOST macro) for restart after a TPWAIT macro. Usually the POST entry is followed sometime later by a RESM entry. However, when a TPPOST precedes the TPWAIT, the TPWAIT returns to the caller immediately without waiting. Then POST is not followed by RESM.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| POST | I D | 0 | P A B O F | PST ADDRESS | PAB ADDRESS | RETURN ADDRESS | WORK ELEMENT ADDRESS | DVT ADDRESS | RPH ADDRESS |

**Byte (hex)**

> **Contents**

**00–03**   Record ID: C"POST"

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**   0

**06–07**   PAB offset

**08–0B**   PST address

**0C–0F**   PAB address

**10–13**   Address of the issuer of the TPPOST macro

**14–17**   Work element address (from RPHWEA)

**18–1B**   PAB DVT address

**1C–1F**   Address of the RPH being posted

## PROA or PROD entry for Profile Add or Delete

**Entry:  PROA** or **PROD**

**VIT option:**
> NRM

**Event:**  Profile Add or Delete

**VIT processing module:**
> ISTRACNR

**Control is returned to:**
ISTSDCPM

This trace record gives information about PROFILE macroinstruction processing. It is written when a VTAM module issues a PROFILE macroinstruction to add or delete an RDTE profile.

| 0 0 0 0 0 1 2 3 | 0 4 | 0 0 0 5 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 1 1 1 1 8 9 A B C D E F |
|---|---|---|---|---|---|---|---|
| PROA PROD | I D | KEY INDEX OR 0 | INPUT PROFILE ADDRESS | OUTPUT PROFILE ADDRESS | RETURN ADDRESS | RDTE ADDRESS | RESOURCE NAME |

**Byte (hex)**
**Contents**
**00–03** Record ID:
- C"PROA": Add a profile
- C"PROD": Delete a profile

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** Key index into the RDTE profile table or 0. Note that an index value of 0 is valid and represents the first slot in the profile table. The index might also be 0 for PROA entries created when a profile is first associated with an RDTE during SYSDEF processing.

**08–0B** Input profile address
**0C–0F** Output profile address
**10–13** Return address of the module calling the ISTSDCRP module or the address of the module issuing PROFILE macro
**14–17** RDTE address
**18–1F** Resource name

# QREQ entry for queued REQSTORE

**Entry: QREQ**
**VIT option:**
SMS
**Event:** Queued storage request
**VIT processing module:**
ISTRACSM
**Control is returned to:**
ISTORFBQ

This trace record identifies a REQSTORE request that was waiting for one or more buffers and is now satisfied.

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 1 1 1 | 1 1 | 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 3 | 4 5 | 6 7 8 9 A B | C D E F |
| QREQ | I D | C B I D | 0 | | PST ADDRESS | | | BUFFER ADDRESS | | | | 0 | | | NUM BUF REQ | | 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"QREQ"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID index value

**06–07** 0

**08–0B** PST address

**0C–0F** Address of buffer obtained

**10–13** 0

**14–15** Number of buffers requested

**16–1B** 0

**1C–1F** Request parameter header (RPH) address

## QRE2 entry for queued REQSTORE

**Entry: QRE2**
**VIT option:**
    SMS
**Event:** Queued storage request
**VIT processing module:**
    ISTRACSM

This trace record is a continuation of QREQ entry. It is generated when a queued REQSTORE macro is processed for more than one buffer. The number of records generated depends on the number of buffers obtained by the REQSTORE request.

VTAM generates up to 31 QRE2 trace records. Each QRE2 entry contains addresses for up to seven buffers. If more than 218 buffers are requested, the last four bytes (1C–1F) in the last QRE2 entry are set to X'FFFF', indicating that not all of the requested buffers are traced.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| QRE2 | SEVEN WORDS OF BUFFER ADDRESSES | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"QRE2"

**04–1F** Up to seven more words of buffer addresses (padded on the right with zeros)

## QRYL entry for query language

**Entry:** QRYL
**VIT option:**
    MSG
**Event:** Query language error
**VIT processing module:**
    ISTRACOT
**Control is returned to:**
    ISTINCUL

This trace record is generated when a nonzero return code is received from the QRYLANG macro. When the return code and reason code indicate that the MVS message service is not active, no QRYL trace record is generated.

| 0 0 0 0 0 1 2 3 | 0 4 | 0 0 0 5 6 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 1 1 1 1 8 9 A B C D E F |
|---|---|---|---|---|---|---|---|
| QRYL | I D | 0 | LANG QUERY BLOCK ADDRESS | LANG QUERY BLOCK LENGTH | RETURN CODE | REASON CODE | FIRST 8 BYTES OF LANGUAGE NAME USED IN QUERY |

**Byte (hex)**
        **Contents**
**00–03**   Record ID: C"QRYL"
**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07**   0
**08–0B**   Language query block address
**0C–0F**   Language query block length (LQBSIZE)
**10–13**   Return code
**14–17**   Reason code
**18–1F**   First eight bytes of language name used in query (LQBINLNG)

## QUE entry for work element queued to PAB

**Entry:** QUE
**VIT option:**
    PSS
**Event:** Work element queued to PAB
**VIT processing module:**
    ISTRACPS
**Control is returned to:**
    ISTAPCTQ

This trace record shows a work element queued to a PAB to allow another VTAM routine to do further processing with the work element. This entry may be followed shortly by a DSP entry, representing the PSS dispatch of this PAB. However, if the PAB is already running, it might not be dispatched again. In this case, no DSP entry follows the TPQUE.

| 0 0 | 0 | 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 | 0 3 | 0 4 | 0 5 | 0 6 7 | 0 8 9 A B | 0 C D E F | 1 0 1 2 3 | 1 4 5 6 7 | 1 8 9 A B | 1 C D E F | |
| QUE | C B I D | I D | S T A T | F L A G S | PST ADDRESS | PAB ADDRESS | RETURN ADDRESS | WORK ELEMENT ADDRESS | MODULE NAME OR DVT ADDRESS | RPH ADDRESS OR 0 |

**Byte (hex)**

   **Contents**

**00–02**   Record ID: C"QUE"

**03**   Control block ID of work element (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)

**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**   Status

   **Bit**   **Meaning**

   **xx.. ....**
      Type of scheduling request

      **Bit**   **Meaning**
      **00**   TPQUE none
      **01**   TPQUE normal
      **10**   TPQUE delay

   **..x. ....**
      0

   **...1 ....**
      Queue is in last-in-first-out (LIFO) order.

   **...0 ....**
      Queue is in first-in-first-out (FIFO) order.

   **.... 1...**
      Registers are saved in the RPH control block.

   **.... .x..**
      0

   **.... ..1.**
      PAB work element queue gate bit (PABWEQG).

   **.... ...0**
      PAB change bit (PABCHNG). PAB *is not* scheduled to run.

   **.... ...1**
      PAB *is* scheduled to run.

**06**   PAB flag field (PABFLAGS)

   **Bit**   **Meaning**

   **1... ....**
      PAB is unconditionally scheduled.

   **.1.. ....**
      PAB closedown is in progress.

   **..1. ....**
      PAB is synchronous.

   **...1 ....**
      PAB extension is present.

   **.... 1...**
      Do not dequeue work element.

> .... .1..
>> Do not detach the RPH.
> .... ..1.
>> Indicates a very extended PAB.
> .... ...1
>> Indicates a slightly extended PAB.

**07** PAB flag field (PABFLGS1)

> **Bit** **Meaning**
> **1... ....**
>> Switch the PST address of this PAB's major control block to the new PST address contained in DYPNWPST.
> **.1.. ....**
>> This PAB has a data space extension.
> **..1. ....**
>> This PAB's major control block is an FMCB.
> **...1 ....**
>> PAB can be referenced in PSW disable mode.
> **.... 1...**
>> PAB is persistent.
> **.... .1..**
>> APSTERM/APSINIT FMCB during PAB dispatch.
> **.... ..xx**
>> Reserved

**08–0B** PST address

**0C–0F** PAB address

**10–13** Address of the issuer of the TPQUE macro

**14–17** Address of work element to be queued

**18–1B** Module name abbreviation (usually bytes 4, 5, 7, and 8 of the module name) or PAB DVT address (high order bit of X'18' = 0). For an explanation of the module naming convention, see "Module names in internal trace records" on page 21. (The module name might be unavailable if the PAB being scheduled is associated with an address space different from the current one.)

**1C–1F** Request parameter header (RPH) address or 0

---

# QUEN entry for work element queued to any control block

**Entry: QUEN**
**VIT option:**
> PSS

**Event:** Work element queued to any control block
**VIT processing module:**
> ISTRACPS

**Control is returned to:**
> ISTAPCTQ

This trace record shows a work element queued to any control block (not just a PAB) to allow another VTAM routine to further process the work element.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| QUEN | I D | C B I D | 0 | PST ADDRESS OR 0 | QUEUE ADDRESS | RETURN ADDRESS | WORK ELEMENT ADDRESS | 0 | RPH ADDRESS OR 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"QUEN"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID of work element (Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*.)

**06–07** 0

**08–0B** Address of PST or 0

**0C–0F** Address of the control block field to which the work element is queued

**10–13** Address of the issuer of the TPQUE NONE macro

**14–17** Address of work element to be queued

**18–1B** 0

**1C–1F** Request parameter header (RPH) address or 0

# RACR entry for LU 6.2 invocation of RACROUTE

**Entry: RACR**

**VIT option:**
APPC

**Event:** Resource access control

**VIT processing module:**
ISTRACAC

**Control is returned to:**
ISTNSCSI

This trace record shows information about the completion of a RACROUTE macro. This trace record is written during security processing for session establishment.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| RACR | I D | R E Q | T Y P E | 0 | ACEE ADDRESS | HALF SESS ID | 0 | RACR COMPLET CODE | REQUEST RETURN CODE | REQUEST REASON CODE |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"RACR"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** RACROUTE request

**X'01'** AUDIT

|  |  |
|---|---|
| | **X'02'** EXTRACT |
| | **X'03'** LIST |
| | **X'04'** VERIFY |
| **06** | RACROUTE type |
| | For AUDIT: 0 |
| | For EXTRACT, LIST, or VERIFY: |
| | **X'01'** CREATE |
| | **X'02'** DELETE |
| | **X'03'** ENCRYPT |
| | **X'04'** EXTRACT |
| **07** | 0 |
| **08–0B** | Address control environment element (ACEE) address |
| **0C–0F** | Half-session ID |
| **10–13** | 0 |
| **14–17** | RACROUTE completion code |
| | **X'00'** Request successfully completed |
| | **X'04'** Request completed with nonzero return/reason code |
| | **X'08'** RACROUTE failure |
| **18–1B** | Request return code from security management product. See the appropriate manual for your security management product for an explanation of the return/reason codes. |
| **1C–1F** | Request reason code from security management product. See the appropriate manual for your security management product for an explanation of the return/reason codes. |

# RCEx entry for RCE macroinstruction

**Entry:** **RCEA, RCEC, RCED, or RCEF**
**VIT option:**
> NRM

**Event:** RCE macroinstruction
**VIT processing module:**
> ISTRACNR

**Control is returned to:**
> The module that issued the RCE macroinstruction

This trace record contains information about an RCEADD, RCECHG, RCEDEL, or RCEFIND macroinstruction issued by a VTAM module.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| RCEA RCEC RCED RCEF | I D | R T N C D | T Y P E | M O D E | KEY VALUE | RETURN ADDRESS | OUT1 OR NEW TABLE NAME | OUT2 OR NEW TABLE NAME | DATA ADDRESS |

**Byte (hex)**
> **Contents**

| | |
|---|---|
| **00–03** | Record ID: C"RCEA" (RCEADD), C"RCEC" (RCECHG), C"RCED" (RCEDEL), C"RCEF" (RCEFIND) |

04     ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05     Return code

06     Type of RCE entry

07     Mode of RCE entry (in hex)
- 01=Base entity mode
- 02=Reference entity mode

08–0F  Key value (base entity name, base index, or reference index, depending on mode)

10–13  Address of the issuer of the macro

14–17  For RCEADD, RCEDEL, and RCEFIND: Value in RCEPOUT1

For RCECHG: First half of new table name

18–1B  For RCEADD, RCEDEL, and RCEFIND: Value in RCEPOUT2

For RCECHG: Second half of new table name

1C–1F  Data address

## RCM entry for RCM PAB dispatch

**Entry: RCM**
**VIT option:**
    HPR
**Event:** Dispatch of RTP context manager (RCM) PAB
**VIT processing module:**
    ISTITCHR
**Control is returned to:**
    The module that issued INTRACE type (RCM)

This trace record is written by the RTP context manager (RCM) to summarize specific activities.

| 00 01 02 03 | 04 | 05 | 06 | 07 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|
| RCM | ID | 0 | INSTANCE | ISSUING MODULE NAME | RCM BASE EXT | RCM RSR EXT | RUPE OPCODE | RPNCB ADDR | RPH ADDR |

**Byte (hex)**
    **Contents**

00–03  Record ID: C"RCM"

04     ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05     0

06     INTRACE macro instance

07–0B  Issuing module name

0C–0F  RCM base extension

10–13  RCM route setup request (RSR) extension

14–17  RUPE opcode

18–1B  RPNCB address

1C–1F   RPH address

---

# RCV entry for RCV PAB dispatch

**Entry:  RCV**
**VIT option:**
>    HPR
**Event:**  Dispatch of MNPS recovery processing (RCV) PAB
**VIT processing module:**
>    ISTITCHR
**Control is returned to:**
>    ISTRVRR1

This trace record is written when a Recovery PAB (RPAB) is dispatched. The entry contains Recovery PAB control block information.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 | 0 | 0 | 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C | D | E | F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| RCV | ID | 0 | STATUS / FLAGS | RPNCB ADDRESS | RCV FSM | SS FSM | RCM FSM | STATE | FLAG 1 / 0 | SESSION COUNT | RVM ADDRESS | RPH ADDRESS |

**Byte (hex)**
>    **Contents**
**00–03**   Record ID: C"RCV"
**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**   0
**06–07**   RCV status flags

>    **Bit      Meaning**
>    **1... ....**
>        Initial dispatch of this RPAB.
>    **.1.. ....**
>        Crypto sessions are being recovered.
>    **..1. ....**
>        Recovery PAB termination is underway.
>    **.... .1..**
>        The RPAB has completed its recovery processing.
>    **.... ..1.**
>        All data to be recovered has been received from the Recovery Manager PAB.
>    **.... ...1**
>        The RPAB is awaiting the status from RTP before beginning termination of the recovery.

**08–0B**   Address of the RTP NCB being recovered by this RPAB
**0C**   State information for internal recovery PAB processing
**0D**   State information for recovery PAB communication with Session Services
**0E**   State information for recovery PAB communication with RTP Context Manager

0F    Recovery APPC state
10    RCV input flags

**Bit    Meaning**
**1... ....**
    This RPAB is still chained in the Recovery Manager session data
    queue.
**.1.. ....**
    No more data will be coming from the Recovery Manager.
**..1. ....**
    The RPAB should terminate recovery when it is next dispatched.
**.... 1...**
    Session Services was unable to send a response IPS to a prior
    MNPS_SESSINFO(Recover)request.
**.... .1..**
    Session Services was unable to send a response IPS to a prior
    MNPS_SESSINFO(Terminate) request.

**11–13**    0
**14–17**    Number of sessions being recovered
**18–1B**    Address of the Recovery Manager control block (RVM)
**1C–1F**    Request parameter header (RPH) address

# RDSC entry for RUPE discard (Part 1)

**Entry:  RDSC**
**VIT option:**
    PIU
**Event:**  Discarding a RUPE
**VIT processing module:**
    ISTRACOT
**Control is returned to:**
    ISTDLCDI

This trace record is written when VTAM's dependent LU server receives an
erroneous signal either from the VTAM configuration services component or from
the dependent LU requester node across the CPSVRMGR session.

| 0 0 0 0 | 0 4 | 0 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| RDSC | I D | R E A S O N | C O D E | RUPE ADDRESS | FIRST 20 BYTES OF PIU DATA |

**Byte (hex)**
    **Contents**
**00–03**    Record ID: C"RDSC"
**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is
    greater than X'FF'.
**05–07**    Reason code (decimal)
    **1**    The dependent LU server abend recovery routine, ISTDLCRR,
        performs clean-up.

| | |
|---|---|
| **2** | CV X'60' is not found. |
| **3** | Dependent LU requester element is not found. |
| **4** | PU element is not found. |
| **5** | LU element is not found. |
| **6** | Storage shortage has occurred. |
| **7** | Dependent LU requester finite state machine is not valid. |
| **8** | Dependent LU requester finite state machine is pending inactive. |
| **9** | CPSVRMGR session pipe activation failure has occurred. |
| **10** | Unrecognized request unit processing element (RUPE) detected. |
| **11** | Start transaction program (STARTTP) failure has occurred. |
| **12** | CPSVRMGR session pipe has already been deactivated. |
| **13** | PU finite state machine is not valid. |
| **14** | SEND_ENCAP_TP process anchor block (PAB) does not exist. |

**08–0B**   Address of the RUPE being discarded

**0C–1F**   First 20 bytes of PIU data (inbound) or RUPE's RU data (outbound) in the RUPE being discarded

## RDS2 Entry for RUPE Discard (Part 2)

**Entry: RDS2**
**VIT option:**
> PIU

**Event:** Discarding a RUPE
**VIT processing module:**
> ISTRACOT

This trace record is a continuation of the RDSC entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| RDS2 | NEXT 20 BYTES OF PIU DATA | RETURN | MODULE NAME |

**Byte (hex)**
> **Contents**

**00–03**   Record ID: C"RDS2"

**04–17**   Bytes 21 through 40 of the PIU data (inbound) or RUPE's RU data (outbound) in the RUPE being discarded

**18–1B**   Return address

**1C–1F**   The module name that initiates the RUPE discard

# RE entry for RPL exit

**Entry: RE**
**VIT option:**
> API

**Event:** RPL exit
**VIT processing module:**
> ISTRACAP

**Control is returned to:**
ISTAPCSX, ISTAICRX, or ISTAICPT

This trace record contains RPL information passed from VTAM to the application program. If the return code or feedback is nonzero, this entry will be generated whether the API option is in effect or not. It is treated as an exception condition and is always traced if the VIT is active, regardless of the VIT options specified.

Refer to *z/OS Communications Server: SNA Programming* for a description of RPL return codes.

| 0 0 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1 1 1<br>0 1 2 3 | 1 1 1 1<br>4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RE | R E Q T Y | E X T D S | I D | R T N C D | F D B 2 | F D B 3 | RPL ADDRESS | EXIT ADDRESS | RPL AREA | RPL RLEN | CID OR 0 | RPL FDBK2 |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–01 | Record ID: C"RE" |
| 02 | RPL request type (RPLREQ) |

| | |
|---|---|
| **X'15'** | SETLOGON |
| **X'16'** | SIMLOGON |
| **X'17'** | OPNDST |
| **X'19'** | CHANGE |
| **X'1A'** | INQUIRE |
| **X'1B'** | INTRPRET |
| **X'1F'** | CLSDST |
| **X'22'** | SEND |
| **X'23'** | RECEIVE |
| **X'24'** | RESETSR |
| **X'25'** | SESSIONC |
| **X'27'** | SENDCMD |
| **X'28'** | RCVCMD |
| **X'29'** | REQSESS |
| **X'2A'** | OPNSEC |
| **X'2C'** | TERMSESS |

| | |
|---|---|
| 03 | Exit definition (RPLEXTDS) |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | Return code (RPLRTNCD) (Refer to *z/OS Communications Server: SNA Programming*) |
| 06 | Feedback code (RPLFDB2) (Refer to *z/OS Communications Server: SNA Programming*) |
| 07 | Feedback data flag (RPLFDB3) (Refer to *z/OS Communications Server: SNA Programming* under INQUIRE macro with OPTCD=APPSTAT) |
| 08–0B | RPL address |
| 0C–0F | Exit address |
| 10–13 | RPL data area pointer (RPLAREA) |
| 14–17 | Record length (RPLRLEN) |
| 18–1B | CID from NIB or from RPL (RPLARG) or 0 |

**1C–1F**  Second RPL feedback area - Sense code (RPLFDBK2)

# RELS entry for release storage

**Entry:  RELS**
**VIT option:**
  SMS
**Event:**  Release storage
**VIT processing module:**
  ISTRACSM
**Control is returned to:**
  ISTORMBD

This trace record provides the status of each RELSTORE request issued by VTAM components to release fixed-length buffers to one of the predefined buffer pools.

RELSTORE is the complement of REQSTORE. Storage obtained by REQSTORE must be released by RELSTORE. However, because more than one buffer can be obtained and released at a time, there may not be a RELS for every REQS, and there may not be a REQS for each RELS. Use the "next buffer address" in RELS and the "number of buffers" in REQS to help determine the correlation between buffers requested and buffers released.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|
| RELS | ID | CBID | 0 | RTNCD | PST ADDRESS | BUFFER ADDRESS | RETURN ADDRESS | NEXT BUFFER ADDRESS OR 0 | REG 1 | CALLER OF UTILITY OR 0 |

**Byte (hex)**
  **Contents**
**00–03**  Record ID: C"RELS"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  Control block ID index value or 0. A 0 is returned if the buffer is not allocated for a specific control block.

  For the index values and the control block IDs they represent, refer to *z/OS Communications Server: SNA Data Areas Volume 1*.
**06**  0
**07**  Return code
**08–0B**  PST address
**0C–0F**  Address of buffer to be released
**10–13**  Address of the issuer of the RELSTORE macro
**14–17**  Pointer to the next buffer to be released if more than one buffer to be released; otherwise, 0
**18–1B**  Register 1 (normally RPH address)
**1C–1F**  Caller of utility routine, or 0

Address of the utility's caller if the RELSTORE macro was issued from a utility routine; otherwise, 0 if the macro was issued directly by the caller without a utility routine (See bytes 10–13 for the address of the issuer of the RELSTORE macro).

# REML entry for TPREMEL macro

**Entry: REML**
**VIT option:**
>   APPC, HPR
**Event:** Element TPREMELed from a work queue; only for selective TPREMELs
**VIT processing module:**
>   ISTRACAC
**Control is returned to:**
>   Module invoking the INTRACE macro that caused the record to be produced

This trace record is written when an element is dequeued from a work queue and processing begins on the element.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E | 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| REML | ID | 0 | INST | TYPE | ISSUING MODULE NAME | 0 | CBID | WORK ELEMENT INFO FIELD OR 0 | WORK ELEMENT ADDRESS | QUEUE ADDRESS | RPH ADDRESS |

**Byte (hex)**
>   **Contents**
**00–03** Record ID: C"REML"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** REML instance within issuing module
**07–0B** Issuing module name
**0C–0E** 0
**0F** Control block ID
**10–13** Information field or 0:
- AMU ID if work element is an AMU
- CONTROL and QUALIFY values if work element is an LU6.2 RPL
- Control point operation code if work element is a RUPE
- 0 if work element is a UECB or TQE.
- DAPT, IN, OUT, or TIMB if REML is traced for HPR

**14–17** Work element address
**18–1B** Address of the queue from which the work element is being dequeued
**1C–1F** Request parameter header (RPH) address

# REMQ entry for APPC

**Entry: REMQ**
**VIT option:**
>   APPC

**Event:** Element TPDEQueued from a PAB; only for selective TPDEQs
**VIT processing module:**
ISTRACAC
**Control is returned to:**
Module invoking the INTRACE macro that caused the record to be produced

This trace record is written when an element is dequeued from a PAB and processing begins on the element.

| 00 01 02 03 | 04 | 05 | 06 | 07 08 09 0A 0B | 0C | 0D | 0E | 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REMQ | ID | 0 | INST | ISSUING MODULE NAME | CNV | RTR | ERROR | CBID | WORK ELEMENT INFO FIELD OR 0 | WORK ELEMENT ADDRESS | PAB ADDRESS | RPH ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"REMQ"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** REMQ instance within issuing module
**07–0B** Issuing module name
**0C** Conversation FSM state or 0
**0D** Router FSM state or 0
**0E** Error or failure FSM state or 0
**0F** Control block ID
**10–13** Information field or 0
- AMU ID if work element is an AMU
- CONTROL and QUALIFY values if work element is an LU6.2 RPL
- Control point operation code if work element is an RUPE
- 0 if work element is a UECB or TQE

**14–17** Work element address
**18–1B** Address of the PAB from which the work element is being dequeued
**1C–1F** Request parameter header (RPH) address

# REQS entry for request storage (Part 1)

**Entry: REQS**
**VIT option:**
SMS
**Event:** Request storage (Part 1)
**VIT processing module:**
ISTRACSM
**Control is returned to:**
ISTORMBA

This trace record provides the status of each REQSTORE request issued by VTAM components to obtain fixed-length buffers from one of the predefined buffer pools. The storage obtained by REQSTORE is released by RELSTORE.

If this entry is associated with an event failure (that is, the return code is nonzero), this entry will be generated whether the SMS option is in effect or not. It is treated as an exception condition and is always traced, regardless of the VIT options specified, if the VIT is active.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| REQS | | | | ID | CBID | 0 | | PST ADDRESS OR 0 | | | | BUFFER ADDRESS OR 0 | | | | RETURN ADDRESS | | | | NUM BUF REQ | | RC | | REG 1 | | | | NUMBER OF AVAIL. BUFFERS | | | |

**Byte (hex)**

    **Contents**

**00–03** Record ID: C"REQS"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block ID index value or 0. A 0 is returned if the buffer is not allocated for a specific control block or if the buffer allocation failed. For the index values and the control block IDs they represent, refer to *z/OS Communications Server: SNA Data Areas Volume 1*.

**08–0B** Pointer to the PST, or 0

**0C–0F** Address of buffer obtained, or 0 if buffer not obtained

**10–13** Address of the issuer of the REQSTORE macro

**14–15** Number of buffers requested

**16–17** Return code from REQSTORE

    **00** Successful.

    **04** Storage obtained, but slowdown threshold exceeded.

    **06** Storage request queued, but buffer pool has to be expanded before buffers can be allocated.

    **08** The number of buffers requested could not be allocated immediately. Depending on the conditions specified on the REQSTORE request and the state of the buffer pool that is requested, the request may or may not be queued for future processing.

        The request is queued for future processing if one of the following conditions is met:

- The REQSTORE request specifies that it can be queued for future processing regardless of the state of the buffer pool from which buffers are requested.

- The REQSTORE request specifies that it can be queued for future processing if specific conditions are met. These conditions are as follows:

  - Number of buffers already allocated plus the number of buffers for which there are outstanding requests does not exceed the maximum size of the buffer pool.

  - Buffer pool is defined to be expandable.

  - Buffer pool expansion has never failed for the buffer pool or the buffer pool has contracted since it failed to expand.

The request is not queued for future processing if any of the following conditions apply:

- The REQSTORE request specifies that it is not to be queued under any circumstances.
- The REQSTORE request specifies that it is to be queued according to one or more of the conditions described previously in this topic and the specified conditions are not met.

**16** This return code applies only to nonexpandable buffer pools. The number of buffers requested exceeds the maximum number which can be allocated in one request. This return code does not indicate that the number of buffers requested is not available; it indicates that the number of buffers exceeds the maximum allowed in one request.

**18–1B** Register 1 (normally RPH address)

**1C–1F** Number of available buffers in pool after this request is processed.

## REQ2 entry for request storage (Part 2)

**Entry: REQ2**
**VIT option:**
　　SMS
**Event:** Request storage (Part 2)
**VIT processing module:**
　　ISTRACSM

This trace record is a continuation of the REQS entry. It is generated when REQSTORE macro was issued for more than one buffer. The number of records generated depends on the number of buffers obtained by the REQSTORE request.

VTAM generates up to 31 REQ2 trace records. Each REQ2 entry contains addresses for up to seven buffers. If more than 218 buffers are requested, the last four bytes, 1C–1F, in the last REQ2 entry are set to X'FFFF' indicating that not all of the requested buffers are traced.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| REQ2 | SEVEN WORDS OF BUFFER ADDRESSES |

**Byte (hex)**
　　**Contents**
**00–03** Record ID: C"REQ2"
**04–1F** Up to seven more words of buffer addresses (padded on the right with zeros)

## RESM entry for resume event processing

**Entry: RESM**
**VIT option:**
　　PSS
**Event:** Resume event processing
**VIT processing module:**
　　ISTRACPS

**Control is returned to:**
    ISTAPCPD

This trace record gives information about which VTAM process is about to be
redispatched after a TPWAIT. A RESM entry follows a POST, although not every
POST has a RESM. Usually the POST entry is followed sometime later by a RESM
entry. However, when a TPPOST precedes the TPWAIT, the TPWAIT returns to the
caller immediately without waiting. Then POST is not followed by RESM. Use the
RPH address to find the original DSP entry, which may have occurred earlier.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|0 1 2 3|4|5|6 7|8 9 A B|C D E F|0 1 2 3|4 5 6 7|8 9 A B|C D E F|
| RESM | I D | C B I D | F L A G S | PST ADDRESS | PAB ADDRESS | WORK EL ADDR OR NEXT DSP Q LVL | WORK ELEMENT QUEUE | PAB DVT ADDRESS OR NAME | RPH ADDRESS |

**Byte (hex)**
      **Contents**
**00–03**   Record ID: C"RESM"
**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is
      greater than X'FF'.
**05**   Control block ID of work element (Refer to *z/OS Communications Server:
      SNA Diagnosis Vol 1, Techniques and Procedures*.)
**06**   PAB flag field (PABFLAGS)

      **Bit     Meaning**
      **1... ....**
            PAB is unconditionally scheduled.
      **.1.. ....**
            PAB closedown is in progress.
      **..1. ....**
            PAB is synchronous.
      **...1 ....**
            PAB extension is present.
      **.... 1...**
            Do not dequeue work element.
      **.... .1..**
            Do not detach the RPH.
      **.... ..1.**
            Indicates a very extended PAB.
      **.... ...1**
            Indicates a slightly extended PAB.
**07**   PAB flag field (PABFLGS1)

      **Bit     Meaning**
      **1... ....**
            Switch the PST address of this PAB's major control block to the
            new PST address contained in DYPNWPST.
      **.1.. ....**
            This PAB has a data space extension.
      **..1. ....**
            This PAB's major control block is an FMCB.

> **...1 ....**
> > PAB can be referenced in PSW disable mode.
> **.... 1...**
> > PAB is persistent.
> **.... .1..**
> > APSTERM/APSINIT FMCB during PAB dispatch.
> **.... ..xx**
> > 0

**08–0B** PST address
**0C–0F** PAB address
**10–13** Work element address or next dispatchable queue level for a very extended PAB
**14–17** PAB work element queue
**18–1B** PAB DVT address or name
**1C–1F** Request parameter header (RPH) address

---

# RIOx entry for Resume I/O (Part 1)

**Entry: RIOx**
**VIT option:**
> CIO

**Event:** Resume I/O
**VIT processing module:**
> ISTRACCI

**Control is returned to:**
> ISTLLCXR, ISTTSCIE, ISTTSCNY, ISTTSCXR, or ISTTSC8W

This trace record is written when the MVS Resume I/O service is invoked and for HPDT only.

Eventually, an INTx or PCIx entry follows the RIOx trace record. Use the CUA field to correlate the entries. Refer to *z/OS Communications Server: SNA Data Areas Volume 1* for a description of the NCB fields.

For record types with suffix I, X, or T, the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

| 00 01 02 03 | 04 | 05 | 06 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|
| RIOx | ID | STATE | 0 | CUA DEVICE | NCB ADDRESS | FLAG BYTES | CAW | OUTPUT CONTROL AREA (STATUS INFORMATION) |

**Byte (hex)**
> **Contents**

**00–03** Record ID: "RIOx" for XCNCB
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** State
**06–07** 0
**08–0B** Channel device name in EBCDIC (either a CUA or device number)

**0C–0F** NCB address
**10–13** Flag bytes (NCBFLAGS)
**14–17** Virtual channel address word (NCBCAW)
**18–1F** Output control area (status information)

# RIO2 entry for Resume I/O (Part 2)

**Entry: RIO2**
**VIT option:**
 CIO
**Event:** Resume I/O (Part 2)
**VIT processing module:**
 ISTRACCI

This trace record is a continuation of the RIOx entry.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RIO2 | | | | SECP | | 0 | | MODID | | FIRST 16 BYTES OF MPC TRANSMIT BLOCK DATA OR 0 OR DEVICE DEPENDENT DATA | | | | | | | | | | | | | | | | 0 | | | | | |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"RIO2"
**04–05** CPNCB_Index_Word
**06–07** 0
**08–09** Module ID
**0A–19** First 16 bytes of MPC transmit block data, or 0, or device dependent data
**1A–1F** 0

# RQE entry for read queue exit

**Entry: RQE**
**VIT option:**
 CMIP
**Event:** Read-queue exit called by CMIP application program
**VIT processing module:**
 ISTITCCM
**Control is returned to:**
 Module invoking the INTRACE macro that caused the record to be
 produced

This trace record is generated when the read-queue exit of a CMIP application program returns from processing a message. The trace record shows the information that is passed from the MIB controller to the application program and the return code passed from the application program.

For detailed descriptions of the fields, refer to *z/OS Communications Server: CMIP Services and Topology Agent Guide*.

If the return code is not 0, this entry is always traced, regardless of the VIT options specified.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 6 | 8 9 | A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| | | | | | | | | | | |
| RQE | | I D | 0 | R E A S O N | LINK ID | INVOKE ID OR 0 | RETURN CODE | USER DATA | LENGTH OF DATA | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C''RQE''

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06–07** Reason code indicating the reason the read-queue exit is called

**08–0B** Link identifier

**0C–0F** Invoke identifier or 0. The invoke identifier can be used to correlate this VIT entry with the CMIP message string in the buffer trace.

**10–13** Return code from application program

**14–17** User data

**18–1B** Length of data being passed to the read-queue exit

**1C–1F** Request parameter header (RPH) address

# RSCx entry for RSCVSCAN macroinstruction (Part 1)

**Entry: RSC**

**VIT option:**
    HPR

**Event:** Invocation of RSCVSCAN macroinstruction

**VIT processing module:**
    ISTITCHP

**Control is returned to:**
    Module invoking the RSCVSCAN macroinstruction

This trace record is written when the RSCVSCAN macroinstruction is issued. The record displays the input RSCV and the output RSCVs (the expanded, condensed, and essential RSCVs).

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
|---|---|---|---|---|---|---|---|
| RSCI, RSCC, RSCE, OR RSCX | I D | F U N C T I O N | E X P A N D | S C A N | RSCV ADDRESS | RETURN ADDRESS | RSCV |

**Byte (hex)**
**Contents**

**00–03** Record ID:

C"RSCI" for tracing input RSCV
C"RSCC" for tracing condensed RSCV
C"RSCE" for tracing essential RSCV
C"RSCX" for tracing expanded RSCV

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Function:
**1** Expand CV85 before building condensed and essential RSCVs
**2** Expand CV85 only
**3** Do not expand CV85 before building condensed and essential RSCVs
**4** Build essential RSCV
**5** Trace condensed RSCV
**6** Trace essential RSCV

**06** Expand Return code:
**14** CV85 error
**15** CV83 missing
**16** CV46 missing

**07** Scan Return code:
**01** V46 length error
**02** V2B length error
**03** V80 length error
**04** Unknown vector
**05** Hop count not valid
**07** V46 not valid
**08** Subvector not valid
**09** CPNAME breakdown error
**12** CV2B prefix error
**13** No RTP path exists
**17** CV4680 missing
**18** CPNAME breakdown error

**08–0B** RSCV address
**0C–0F** Return address
**10–1F** 16 bytes of RSCV

# RSC2 entry for RSCVSCAN macroinstruction (Part 2)

**Entry: RSC2**
**VIT option:**
HPR

**Event:** Invocation of RSCVSCAN macroinstruction
**VIT processing module:**
    ISTITCHP

This trace record is a continuation of the RSCx entry.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| RSC2 | | | | 28 BYTES OF RSCV | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"RSC2"
**04–1F**  28 bytes of RSCV

# RTP entry for RTP PAB dispatch

**Entry:  RTP**
**VIT option:**
    HPR
**Event:**  Dispatch of a rapid transport protocol (RTP) PAB
**VIT processing module:**
    ISTITCHR
**Control is returned to:**
    ISTRPCRT

This trace record is written when an RTP PAB is dispatched. The entry contains
RTP state and adaptive rate-base congestion control (ARB) information.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| RTP | | | | I D | 0 | P A T H S W I T C H | A R B F L A G S | LAST BYTE SENT | | | | LAST BYTE RCVD | | | | R T P C O N N S T | T I M E R S | N U M B E R O F | N L P S | NUMBER OF BYTES TO SEND | | | | RPNCB ADDR | | | | RPH ADDR | | | |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"RTP"
**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is
        greater than X'FF'.
**05**    0
**06**    RTP path switch fields:

> **Bit** **Meaning**
> **1... ....**
> > Path switch—Waiting for LU-LU session services or topology and routing services (RPN_WAIT_PSWCH)
>
> **.1.. ....**
> > Path switch—Waiting for partner (RPN_WAIT_PS_STAT)
>
> **..1. ....**
> > Back pressure applied (RPN_BACK_PRESSURE)
>
> **...1 ....**
> > Status pending (RPN_STATUS_PENDING)
>
> **.... 1...**
> > Last message bit set in the NLP transport header (RPN_LAST_MSG)
>
> **.... .1..**
> > Path switch in progress (RPN_PSWCH_STATE)
>
> **.... ..1.**
> > Modify RTP command in progress (RPN_MRTP)
>
> **.... ...1**
> > Allow reporting of lost data (RPN_REPORT_GAPS)

**07**   ARB Flags:

> **Bit** **Meaning**
> **1... ....**
> > Echo pending (RPN_ECHO_PENDING)
>
> **.1.. ....**
> > GAP pending (RPN_GAP_PENDING)
>
> **..xx ....**
> > Mode (ARB_MODE):
> > **00**   Green
> > **01**   Yellow
> > **10**   Red
>
> **.... xxxx**
> > Number retries (RPN_NUM_RETRY)
> >
> > (Retransmitted data)

**08–0B**   Last byte sequence number sent (RPN_NEXT_BYTE_XMIT)
**0C–0F**   Last byte sequence number received (RPN_LAST_BYTE_RCV)
**10**   RTP connection state (RPN_CONN_STATE)
**11**   Timers:

> **Bit** **Meaning**
> **1... ....**
> > Burst timer expired.
>
> **.1.. ....**
> > Short request timer expired.
>
> **..1. ....**
> > Liveness timer expired.
>
> **...1 ....**
> > HPRPST path switch timer expired.
>
> **.... 1...**
> > Refifo timer expired.
>
> **.... .1..**
> > New route timer for path switch expired.
>
> **.... ..xx**
> > Unused; available

**12–13**   Number of retransmitted NLPs (RPN_REXMITTED_NLPS)

| | 14–17 | Number of bytes to send in the current burst interval (RPN_BYTES_TO_SEND) |

**14–17** Number of bytes to send in the current burst interval (RPN_BYTES_TO_SEND)
**18–1B** RPNCB address
**1C–1F** Request parameter header (RPH) address

## RTP2 entry for RTP PAB dispatch (Part 2)

**Entry: RTP2**
**VIT option:**
> HPR

**Event:** Dispatch of a rapid transport protocol (RTP) PAB
**VIT processing module:**
> ISTITCHR

This trace record is a continuation of the RTP entry.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTP2 | | | | RTP FLAGS2 | | MNPS | | FLAGS | | STATUS | | XMIT STATUS | | RECEIVED | | ECHO RECEIVED | | 0 | | RPNCB EXT ADDRESS | | | | INBOUND HIGH NLP NUMBER | | | | PENDING OK-TO-ACK SEQ NUMBER | | 0 | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"RTP2"
**04–05** RTP flags

> Byte 1

> **Bit** **Meaning**
> **1... ....**
>> This RTP uses a CoS of CPSVCMG.
> **.1.. ....**
>> This RTP uses a CoS of RSETUP.
> **..1. ....**
>> Remote node is a network node (valid only for CPSVCMG RTPs).
> **...1 ....**
>> Remote node is in this native subnet (valid only for CPSVCMG RTPs).
> **.... 1...**
>> Remote node is border node (valid only for CPSVCMG RTPs).
> **.... .11.**
>> Component initiating pipe termination.
> **.... ...1**
>> Nonmobile endpoint is performing a last ditch path switch attempt.

> Byte 2

> **Bit** **Meaning**

**1... ....**
    Waiting for Route_Setup reply during path switch state

**.xx. ....**
    Adaptive Rate Based Algorithm Used
    **00**      ARB Mode Algorithm
    **01**      ARB Responsive Mode Algorithm

**...x x...**
    Unused; available

**.... .xxx**
    Backpressure reason code (valid only when
    RPN_BACK_PRESSURE is on. See byte 6, bit 3 in the RTP entry)

    **B'001'**  Path switch in progress

    **B'010'**  Send queue backed up (congested)

    **B'011'**  Storage shortage

    **B'100'**  Stalled RTP pipe

    **B'101'**  Wait-for-acknowledgement queue maximum reached

**06–07**  Multinode Persistent Sessions (MNPS) flags:

Byte 1

**Bit**    **Meaning**

**1... ....**
    This end of the RTP connection is associated with an MNPS
    application.

**.1.. ....**
    The partner endpoint is associated with an MNPS application.

**..1. ....**
    MNPS endpoint path switch is in progress.

**...1 ....**
    Incoming data should be discarded until MNPS recovery is
    complete.

**.... 1...**
    MNPS better path path switch processing is in progress.

**.... .1..**
    This RTP connection maintains a real connection path that is
    different from its computed session path.

**.... ..1.**
    MNPS Coupling Facility structure data needs to be repopulated.

Byte 2

**Bit**    **Meaning**

**1... ....**
    MNPS recovery is underway.

**..1. ....**
    The recovery PAB has given permission to delete the RPNCB.

**...1 ....**
    RTP Context Manager has informed RPAB that the RPNCB must
    be deleted.

**.... 1...**
    Recovery is being terminated, but RCM is waiting for completion
    of ALS processing before continuing.

**.... .1..**
    This recovery PAB is chained on the ATCVT list of recovery PABs.

> **.... ..1.**
>> PU Termination processing is waiting for RTP Connection cleanup.
> **.... ...1**
>> Route Setup processing is required during RTP recovery.

**08–09** Last status number transmitted

**0A–0B** Last status number received

**0C–0D**

Last status number echoed correctly by partner

**0E–0F** 0

**10–13** Pointer to MNPS RPNCB extension

**14–17** Highest contiguous inbound NLP sequence number acknowledged by this endpoint

**18–1B** Sequence number associated with first NLP that is pending notification from TSC that the data can be acknowledged to partner endpoint

**1C–1F** 0

---

# RTPE entry for RTP error detection

**Entry: RTPE**
**VIT option:**
> HPR

**Event:** Error detected by rapid-transport protocol (RTP)
**VIT processing module:**
> ISTITCHR

**Control is returned to:**
> ISTRPCRC or ISTRPCRR

This trace record is written when rapid-transport protocol (RTP) detects an error condition, including protocol violations.

| 00 01 02 03 | 04 | 05 | 06 | 07 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 | 15 | 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RTPE | ID | 0 | INSTANCE | ISSUING MODULE NAME | TSCB ADDR | SENSE CODE | ACTION | IORO | 0 | RPNCB ADDR | RPH ADDR |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"RTPE"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Instance of the trace in the issuing module

**07–0B** Name of the module that issued this trace entry

**0C–0F** TSCB address containing network layer packet (NLP)

**10–13** Sense code indicating error condition (MRPFA sense code)

**14** Action:

> **01** RTP termination

> **02** Packet discard

|  |  |
|---|---|
| **03** | Packet processed (error ignored) |
| **15** | |
| | **I—Inbound** |
| | Partner error detected |
| | **O—Outbound** |
| | Origin node error detected |
| **16–17** | 0 |
| **18–1B** | RPNCB address |
| **1C–1F** | Request parameter header (RPH) address |

# RTPP entry for path switch processing

**Entry: RTPP**
**VIT option:**
> HPR

**Event:** Path switch
**VIT processing module:**
> ISTITCHR

**Control is returned to:**
> Module invoking the INTRACE macro that caused the record to be produced

This trace record is generated at various points during path switch processing.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTPP | | | | ASID | PS EVENT | PATH SW 1 | PATH SW 2 | PATH SW 3 | PS FLAVOR | NUM PSRY | RTP CONST | NRTEO I WUMTE ER | PST AW I T IM HTE CR H | | | RETURN ADDRESS | | | | RUPE OPCODE | | | | RPNCB ADDRESS | | | | RPH ADDRESS | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"RTPP"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** If nonzero, indicates one of the following path switch events:
> C - Path switch successfully completed
> F - Path switch failed
> O - Path switch overridden by partner
> S - Path switch started

**06** RTPP path switch 1 fields:

> **Bit     Meaning**
> **1... ....**
> > Path switch—Waiting for new route (RPN_WAIT_PS_APPN)
> **.1.. ....**
> > Path Switch—Waiting for partner (RPN_WAIT_PS_STAT)
> **..1. ....**
> > Path switch needed (RPN_PS_NEEDED)
> **...1 ....**
> > Path switch in progress (RPN_PSWCH_STATE)

> > **.... 1...**
> > > Modify RTP command in progress (RPN_MRTP)
> > **.... .1..**
> > > RPN_RS_OUTSTANDING
> > **.... ..1.**
> > > RPN_INTERNET
> > **.... ...1**
> > > RPN_LAST_DITCH

**07**   RTPP path switch 2 fields:

> **Bit**   **Meaning**
> **1... ....**
> > RPN_PS_TIMER
> **.1.. ....**
> > RPN_NR_TIMER
> **..xx x...**
> > RPN_PS_REASON
> **.... .x..**
> > RPN_PSW_STARTED_MSG
> **.... ..xx**
> > RPN_CONN_TYPE

**08**   RTPP path switch 3 fields:

> **Bit**   **Meaning**
> **1... ....**
> > RPN_LOCAL_MNPS
> **.1.. ....**
> > RPN_REMOTE_MNPS
> **..1. ....**
> > RPN_TWOSTEP
> **...1 ....**
> > RPN_ENDPOINT_PS
> **.... xxxx**
> > Unused; available

**09**   RPN_PS_FLAVOR
**0A**   RPN_NUM_PS_RETRY
**0B**   RPN_CONN_STATE
**0C–0D**
> RPN_NEW_ROUTE_T (low half)
**0E–0F**   RPN_PATHSWITCH_T (low half)
**10–13**   Return address
**14–17**   RUPE opcode
**18–1B**   RPNCB address
**1C–1F**   Request parameter header (RPH) address

---

# RTSx entry for route setup (Part 1)

**Entry:  RTS**
**VIT option:**
> HPR
**Event:**  Route setup sent or received
**VIT processing module:**
> ISTITCHR
**Control is returned to:**
> ISTTSCAR, ISTTSCWS, or ISTTSC9S

This trace record is written when a FID2 route-setup record is sent or received at the DLC layer. The RTSI and RTSO entries provide, respectively, inbound and outbound information about the FID2 rapid transport protocol (RTP) route-setup signal when it is detected at the DLC layer.

| 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D | E F |
| RTSI OR RTSO | | ID · 0 | CBID · ISSUING | MODULE NAME | | TSCB ADDR | 0 | | TSCFLG1 · TSCFLG2 | TSCFLG3 · TSCFLG4 | TSCFLG5 · 0 | LENGTH | | RPH ADDR | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–03 | Record ID:<br>C"RTSI" for inbound<br>C"RTSO" for outbound |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06 | Control block ID |
| 07–0B | Name of the module that issued this trace entry |
| 0C–0F | TSCB address |
| 10–13 | 0 |
| 14 | TSCFLG1 |
| 15 | TSCFLG2 |
| 16 | TSCFLG3 |
| 17 | TSCFLG4 |
| 18 | TSCFLG5 |
| 19 | TSCB FID type |
| 1A–1B | Length of data (TSCILNG) |
| 1C–1F | Request parameter header (RPH) address |

# RTS2 entry for route setup (Part 2)

**Entry:  RTS2**

**VIT option:**
> HPR

**Event:**  Route setup sent or received

**VIT processing module:**
> ISTITCHR

This trace record is a continuation of the RTS entry.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RTS2 | | | | 28 BYTES OF FID2 ROUTE SETUP | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"RTS2"

**04–1F**  28 bytes of the FID2 route setup

# RVM entry for RVM PAB dispatch

**Entry: RVM**

**VIT option:**

   HPR

**Event:** Dispatch of recovery manager (RVM) PAB

**VIT processing module:**

   ISTITCHR

**Control is returned to:**

   ISTRVMRT

This trace record is written when a Recovery Manager PAB is dispatched. The entry contains Recovery Manager control block information.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RVM | | | | I D | 0 | F S M | F F S T I D | STATUS FLAGS | | | | S T A T U S | 0 | | | F L A G S | WORK ELT COUNT | | R P A B S | W O R K I N G | R P A B S | C O M P L E T E | 0 | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"RVM"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05**  0

**06**  Recovery Manager state

**07**  Identifier of last FFST probe issued by this instance of Recovery Manager

**08–0B**  Recovery Manager status flags

**Bit**  **Meaning**

**1... ....**

   Another VTAM has conceded ownership of the ASB to this VTAM as a result of a takeover.

**.1.. ....**

   This VTAM has claimed ownership of the ASB.

**..1. ....**
    Recovery Manager is terminating.

**...1 ....**
    Recovery Manager has processed the request to begin termination that was sent by Configuration Services.

**.... 1...**
    Recovery Manager has incremented the ACDEB "pending recovery" session count.

**.... .1..**
    All the Recovery PABs managed by this Recovery Manager instance have completed.

**.... ..1.**
    Recovery Manager initialization is continuing.

**.... ...1**
    The DSDCB for this Recovery Manager has been freed.

**09**

| Bit | Meaning |
|-----|---------|

**1... ....**
    The takeover MNSPL for this Recovery Manager has been freed.

**.1.. ....**
    At least one set of session information blocks was incomplete.

**..1. ....**
    No APPC LME data was available for this application.

**...1 11..**
    ENCRTYPE encoding for this application.

**.... ..1.**
    The forced takeover request sent for this application has been accepted by the current owning node.

**.... ...X**
    0

**0A**

| Bit | Meaning |
|-----|---------|

**1... ....**
    Application is also a member of a generic resource.

**.1.. ....**
    Need to send an XCF takeover request.

**..1. ....**
    Takeover reply has been received.

**...1 ....**
    APPC list reading is required.

**.... 1...**
    This takeover request is a forced takeover.

**.... .XXX**
    0

**0B**    0
**0C**    Planned takeover status
**0D–0F**  0
**10**    Recovery Manager interface flags

| Bit | Meaning |
|-----|---------|

**1... ....**
Recovery Manager is still accepting work to do from other components.

**.1.. ....**
One or more Recovery PABs have completed work.

**..1. ....**
Recovery Manager should terminate recovery processing at the next dispatch.

**...1 ....**
Initial dispatch of PAB.

**.... XXXX**
0

**11–13** Number of work elements queued to Recovery Manager
**14–15** Number of Recovery PABs (RPABs) that are in the process of recovery
**16–17** Number of Recovery PABs (RPABs) that have completed their recovery processing
**18–1B** 0
**1C–1F** Request parameter header (RPH) address

# RXMT entry for adding PDU to RPNCB queue

**Entry: RXMT**
**VIT option:**
HPR
**Event:** A PDU is transmitted.
**VIT processing module:**
ISTITCHR
**Control is returned to:**
ISTRPCRX

This trace record is written when a PDU is added to the RPNCB queue for data waiting to be retransmitted.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| RXMT | I 0 N F<br>D A L<br>B A<br>S G<br>S | ISTSND<br>ADDRESS | SEGMENT<br>ADDRESS<br>OR 0 | PIU<br>LENGTH | SEQUENCE<br>NUMBER | NCB<br>ADDRESS | RPH<br>ADDRESS |

**Byte (hex)**

**Contents**
**00–03** Record ID: C"RXMT"
**04** Primary address space ID
**05** 0
**06** Number of acknowledged bytes
**07** Flags

**Bit     Meaning**
**x... ....**
Available

```
.1.. ....
```
　　　　Outbound CV29 processed
```
..1. ....
```
　　　　Last message indicator
```
...0 0...
```
　　　　First transmission
```
...0 1...
```
　　　　Retransmit
```
...1 0...
```
　　　　Retransmitted
```
.... .1..
```
　　　　TSOP failed to get storage for token.
```
.... ..1.
```
　　　　Need to free RPN_OUTBOUND_CV29_Q entry.
```
.... ...1
```
　　　　Send was built due to a MNPS recovery.

**08–0B**　ISTSND address
**0C–0F**　Segmentation address or 0
**10–13**　PIU length
**14–17**　Sequence number
**18–1B**　NCB address
**1C–1F**　RPH address

## SBAL entry for Storage Block Address Lists (Part 1)

**Entry: SBAL**
**VIT option:**
　　　CIA
**Event:** Capture SBAL
**VIT processing module:**
　　　ISTITCOD
**Control is returned to**
　　　ISTLLCIE, ISTLLCWC, ISTLLCWI

This trace record is written when the OSA-Express QDIO or HiperSockets Storage Block Address List is altered. Specifically, SBAL records will follow or be preceded by an SLSB record. The SLSB shows the contents of all 128 status bytes while the SBAL record shows the contents of a single SBAL.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| SBAL | | | | ID | | 0 | | INDEX | SBFLAG 0 | SBFLAG 1 | SBFLAG 14 | SBFLAG 15 | | | | SBALE 1 | | | | SBALE 2 | | | | SBALE 3 | | | | SBALE 4 | | RPH ADDRESS | |

**Byte (hex)**
　　　**Contents**
**00–03**　Record ID: C"SBAL"

**04**      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–06**    0

**07**      SBAL index

**08–0B**    SBALE_SBALF_0, SBALE_SBALF_1, SBALE_SBALF_14, and SBALE_SBALF_15 respectively.

**Note:** Nonzero SBALE_SBALF_14 indicates SBAL error.

**0C–0F**    Contents of first SBALE:

Format:

**Bit 0**     SBALE_Last_Entry

**Bit 1**     SBALE_Next_Contig

**Bits 2–3**

        SBALE–FT (fragment type)

        **"00"B**    Not fragmented

        **"01"B**    First fragment

        **"10"B**    Middle fragment

        **"11"B**    Last fragment

**Bits 4–19**

        SBALE_length

**Bits 20–31**

        SBALE_Addr page offset

**10–13**    Contents of continuation SBALEs (if applicable).

**14–17**    Contents of continuation SBALEs (if applicable).

**18–1B**    Contents of continuation SBALEs (if applicable).

**1C–1F**    Request parameter header (RPH) address.

# SBA2 entry for Storage Block Address Lists (Part 2)

**Entry: SBA2**
**VIT option:**
     CIA
**Event:** Capture SBAL
**VIT processing module:**
     ISTITCOD

This trace record is written subsequent to SBAL or SBA2 entries when the previous entry did not contain enough room to capture up to and including the last active SBALE.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| S B A 2 | S B A L E X | S B A L E X + 1 | S B A L E X + 2 | S B A L E X + 3 | S B A L E X + 4 | S B A L E X + 5 | S B A L E X + 6 |

**Byte (hex)**

**Contents**

00–03   Record ID: C"SBA2"

04–07   Contents of continuation SBALEs (see SBAL record for SBALE format)

08–0B   Contents of continuation SBALEs (if applicable)

0C–0F   Contents of continuation SBALEs (if applicable)

10–13   Contents of continuation SBALEs (if applicable)

14–17   Contents of continuation SBALEs (if applicable)

18–1F   Contents of continuation SBALEs (if applicable)

## SCHD entry for schedule of PAB dispatch

**Entry:  SCHD**

**VIT option:**
    PSS

**Event:**  Schedule of PAB dispatch

**VIT processing module:**
    ISTRACPS

**Control is returned to:**
    ISTAPCTS

This trace record shows the schedule of a PAB to be dispatched and the last work element, if any, that was queued to the PAB. This entry should be followed shortly by a DSP entry, representing the PSS dispatch of this PAB.

| 0 0 0 0 | 04 | 05 | 06 0 7 | 0 0 0 0 8 9 A B | 0 0 0 0 C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | | | | | | | | | |
| SCHD | I D | S T A T | F L A G S | PST ADDRESS | PAB ADDRESS | RETURN ADDRESS | WORK EL Q OR NEXT DSP Q LVL | MODULE NAME OR DVT ADDRESS | RPH ADDRESS |

**Byte (hex)**

**Contents**

00–03   Record ID: C"SCHD"

04        ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05        Status

| Bit | Meaning |
|---|---|
| **xx.. ....** | 0 |
| **..1. ....** | TPSCHED unconditional |
| **...x ....** | 0 |
| **.... 1...** | Registers are saved in the RPH control block. |
| **.... .x..** | 0 |
| **.... ..1.** | PAB work element queue gate bit (PABWEQG). |
| **.... ...0** | PAB change bit (PABCHNG) = 0 if the PAB *is not* scheduled to run. |
| **.... ...1** | PABCHNG=1 if the PAB *is* scheduled to run. |

**06**   PAB flag field (PABFLAGS)

| Bit | Meaning |
|---|---|
| **1... ....** | PAB is unconditionally scheduled. |
| **.1.. ....** | PAB closedown is in progress. |
| **..1. ....** | PAB is synchronous. |
| **...1 ....** | PAB extension is present. |
| **.... 1...** | Do not dequeue work element. |
| **.... .1..** | Do not detach the RPH. |
| **.... ..1.** | Indicates a very extended PAB. |
| **.... ...1** | Indicates a slightly extended PAB. |

**07**   PAB flag field (PABFLGS1)

| Bit | Meaning |
|---|---|
| **1... ....** | Switch the PST address of this PAB's major control block to the new PST address contained in DYPNWPST. |
| **.1.. ....** | This PAB has a data space extension. |
| **..1. ....** | This PAB's major control block is an FMCB. |
| **...1 ....** | PAB can be referenced in PSW disable mode. |
| **.... 1...** | PAB is persistent. |
| **.... .1..** | APSTERM/APSINIT FMCB during PAB dispatch. |
| **.... ..xx** | 0 |

**08–0B**   PST address associated with PAB

**0C–0F**   PAB address

**10–13**   Address of the issuer of the TPSCHED macro

**14–17**  PAB work element queue or next dispatchable queue level for a very extended PAB

**18–1B**  Module name abbreviation (usually bytes 4, 5, 7, and 8 of the module name) or PAB DVT address (high order bit of X'18' = 0). For an explanation of the module naming convention, see "Module names in internal trace records" on page 21. The module name might be unavailable if the PAB being scheduled is associated with an address space different from the current one.

**1C–1F**  Request parameter header (RPH) address

---

# SIGA entry for Signal Adapter event

**Entry:  SIGA**
**VIT option:**
    CIO
**Event:**  Signal Adapter instruction
**VIT processing module:**
    ISTITCOD
**Control is returned to**
    ISTLLCIE, ISTLLCRB, ISTLLCWC, ISTLLCWI

This trace record is written immediately subsequent to the issuance of the Signal Adapter instruction. The Signal Adapter instruction is primarily used to notify the OSA-Express QDIO or HiperSockets adapter that the status of one or more SBALs has changed (for example: data is available for a write operation). SIGA is similar in function to SIO/SSCH. SIGA is also used to communicate with the VM operating system when z/OS is executing in a specific type of virtual machine.

| 00 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 | 16 17 | 18 19 | 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIGA | ID | 0 | FUNCTION | COND CODE | DEVICE ADDRESS | NCB ADDRESS | SIW | QUEUE INDEX 1 | QUEUE INDEX 2 | 0 | MODULE IDENTIFIER | RPH ADDRESS |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"SIGA"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  0
**06**  SIGA function:
- C"R" for SIGA-r (SIGA read)
- C"S" for SIGA-s (SIGA synchronize)
- C"W" for SIGA-w (SIGA write)

**07**  Left nibble: Condition code from the SIGA instruction.

Right nibble: Program mask.
**08–0B**  Device address in EBCDIC (as specified in the TRL deck)

**0C–0F** DINCB address
**10–13** Subsystem Identification Word: X'0001' followed by the Subchannel Number that corresponds to the device address.
**14–15** Queue indices bit mask (1):
- SIGA read: Specifies read queues for which the Storage List Status Bytes are to be refreshed.
- SIGA write: Specifies write queues for which the Storage List Status Bytes have been updated.
- SIGA synch: Specifies write queues for which the Storage List Status Bytes are to be refreshed.

**16–17** Queue indices bit mask (2):
- SIGA read: X'0000'
- SIGA write: X'0000'
- SIGA synch: Specifies read queues for which the Storage List Status Bytes are to be refreshed.

**18–19** 0
**1A–1B** Module identifier for the module that issued the INTRACE
**1C–1F** Request parameter header (RPH) address

# SIOx entry for Start I/O

**Entry: SIOD, SIOI, SIOL, SIOT, or SIOX**
**VIT option:**
CIO
**Event:** Start I/O
**VIT processing module:**
ISTRACCI
**Control is returned to:**
ISTTSCIO or ISTTSCLC

This trace record is written when a Start I/O SVC is issued.
- SIOD is generated for OSA-Express QDIO or HiperSockets adapters.
- SIOI is generated for communication controllers and local SNA cluster controllers.
- SIOL is generated for local non-SNA cluster controllers.
- SIOT is generated for a TCP/IP DLC connection.
- SIOX is generated for channel-to-channel-attached hosts and channel-attached IBM 3172 Interconnect Controllers.

Eventually, an INTx or PCIx entry follows the SIO trace record. Use the CUA field to correlate the entries.

Refer to *z/OS Communications Server: SNA Data Areas Volume 1* for a description of the NCB fields.

For record types with suffix D, I, X, or T, the CIO events are also captured within the NCB (pointed to by NCBCIOMV). The NCB trace table is mapped by NCBCIOAR.

## SIOD and SIOT mapping and field descriptions

| 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |

| SIOT | I D | S T A T E | 0 | T Y P E | CUA DEVICE | NCB ADDRESS | FLAG BYTES | CAW | CCW |
|---|---|---|---|---|---|---|---|---|---|

**Byte (hex)**

**Contents**

**00–03** Record ID:

"SIOD" for DINCB

"SIOT" for RWNCB

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** State

**06** 0

**07** For SIOD, 0

For SIOT, type:

- For the CDLC DLC: C

- For the HYPERchannel DLC: H

- For the LCS DLC: L

- For the CLAW DLC: W

- For the CTC DLC: X

- For the Hyperchannel DLC: H

**08–0B** Channel device name in EBCDIC (either a CUA or device number)

**0C–0F** NCB address

**10–13** Flag bytes (NCBFLAGS)

**14–17** Virtual channel address word (NCBCAW)

**18–1F** First channel command word (CCW)

# SIOI mapping and field descriptions

| 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |

| SIOI | I D | S T A T E | W R I T E | 0 | CUA DEVICE | NCB ADDRESS | FLAG BYTES | CAW | FIRST WRITE CCW ADDRESS OR 0 | R E A D | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|

**Byte (hex)**

**Contents**

**00–03** Record ID:

- "SIOI" for ICNCB

| | |
|---|---|
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | Link station state |
| **06** | Write start operation code |
| **07** | 0 |
| **08–0B** | Channel device name in EBCDIC (either a CUA or device number) |
| **0C–0F** | NCB address |
| **10–13** | Flag bytes (NCBFLAGS) |
| **14–17** | Virtual channel address word (NCBCAW) |
| **18–1B** | If link is active, first WRITE CCW and READ START operation code (06 will be WRITE START operation code), or 0 |
| | If link is not active, first channel command word (CCW) |
| **1C** | Read start operation code |
| **1D–1F** | 0 |

## SIOL mapping and field descriptions

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 | 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C | D E F |
| SIOL | | I D | S T A T E | 0 | CUA DEVICE | NCB ADDRESS | FLAG BYTES | CAW | FIRST WORD OF SELECT CCW OR 0 | C O D E | 0 |

**Byte (hex)**

    **Contents**

| | |
|---|---|
| **00–03** | Record ID: "SIOL" for LDNCB |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | State |
| **06–07** | 0 |
| **08–0B** | Channel device name in EBCDIC (either a CUA or device number) |
| **0C–0F** | NCB address |
| **10–13** | Flag bytes (NCBFLAGS) |
| **14–17** | Virtual channel address word (NCBCAW) |
| **18–1B** | If link is active and select operation; CCW and operation code, or 0. If link is not active, first channel command word (CCW) |
| **1C** | Code |
| **1D–1F** | 0 |

## SIOX mapping and field descriptions

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIOX | | | | ID | STATE | 0 | | CUA DEVICE | | | | NCB ADDRESS | | | | FLAG BYTES | | | | CAW | | | | OUTPUT CONTROL AREA (STATUS INFORMATION) | | | | | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: "SIOX" for XCNCB

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** State

**06–07** 0

**08–0B** Channel device name in EBCDIC (either a CUA or device number)

**0C–0F** NCB address

**10–13** Flag bytes (NCBFLAGS)

**14–17** Virtual channel address word (NCBCAW)

**18–1F** Output control area (status information)

# SIO2 entry for Start I/O (Part 2)

**Entry:  SIO2**

**VIT option:**
>    CIO

**Event:**  Start I/O (Part 2)

**VIT processing module:**
>    ISTRACCI

This trace record is a continuation of the SIOI, SIOL, SIOT, and SIOX entries.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIO2 | | | | SECP | | 0 | | MODID | | FIRST 16 BYTES OF MPC TRANSMIT BLOCK DATA OR 0 OR DEVICE DEPENDENT DATA | | | | | | | | | | | | | | | | 0 | | | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"SIO2"

**04–05** CPNCB_Index_Word

**06–07** 0

**08–09** Module ID

**0A–19** First 16 bytes of MPC transmit block data, or 0, or device dependent data

**1A–1F** 0

# SLSB entry for OSA-Express QDIO or HiperSockets Storage List Status Bytes (Part 1)

**Entry:** SLSB
**VIT option:**
    CIA
**Event:** Capture SLSBs
**VIT processing module:**
    ISTITCOD
**Control is returned to**
    ISTLLCIE, ISTLLCWC, ISTLLCWI, ISTITCOD

This trace record is written when the OSA-Express QDIO or HiperSockets Storage List Status Bytes are altered. Specifically, this record is written:

- Upon completion of the construction of write structures for a given write queue and immediately preceding the issuance of the Signal Adapter instruction.
- Upon detection of a write completion during OSA-Express QDIO or HiperSockets timer processing.
- Upon read completion (PCI interrupt processing).

      Prior to read completion processing

  **Note:** The PCID record is also used to capture the SLSBs. If the PCID record captures all 128 status bytes, the SLSB record is skipped. If the PCID record does not capture all 128 status bytes, the SLSB record does not start where PCID left off but instead replicates the count/status pairs in the PCID.

- Upon a priority queue entering a congested state.

| 00 | 01 02 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SLSB | | ID | 0 | CONGESTION STATE | PRIORITY | DEVICE ADDRESS | NCB ADDRESS | INDEX NEXT | INDEX LAST | COUNT | STATUS | COUNT | STATUS | COUNT | STATUS | COUNT | STATUS | COUNT | STATUS | RPH ADDRESS |

**Byte (hex)**
    **Contents**
**00–03**    Record ID: C"SLSB"
**04**    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**    0
**06**    Congestion state; the value X'04' indicates that a priority queue has entered a congested state.
**07**    Queue priority
**08–0B**    Device address in EBCDIC (as specified in the TRL deck).

**0C–0F**  DINCB address
**10**  Contents of DINCB_Q_Data_Index_Next
- Next Empty Read for read SLSBs
- Next Write to Complete for write SLSBs

**11**  Contents of DINCB_Q_Data_Index_Last
- Last Empty Read for read SLSBs
- Last Write to Complete for write SLSBs

**12–13**  First of up to 128 count/status pairs

Starting at SLSB(1), these fields represent the number of consecutive SLSBs containing the same status, and what the status value is. See SLS_SLSB_Status constants for status values.

**14–15**  Continuation count/status pairs (if applicable)
**16–17**  Continuation count/status pairs (if applicable)
**18–19**  Continuation count/status pairs (if applicable)
**1A–1B**  Continuation count/status pairs (if applicable)
**1C–1F**  Request parameter header (RPH) address

## SLS2 entry for OSA-Express QDIO or HiperSockets Storage List Status Bytes (Part 2)

**Entry:  SLS2**
**VIT option:**
    CIA
**Event:**  Capture SLSBs
**VIT processing module:**
    ISTITCOD

This trace record is written when the previous SLSB or SLS2 entry did not contain enough room for all the count/status pairs needed to represent the entire SLSB.

| 00 | 04 | 08 | 0C | 10 | 14 | 18 | 1C |
|----|----|----|----|----|----|----|----|
| SLS2 | Count Status | Count Status | Count Status | Count Status | Count Status | Count Status | Count Status |

**Byte (hex)**
    **Contents**
**00–03**  Record ID: C"SLS2"
**04–05**  Continuation count/status pairs
**06–07**  Continuation count/status pairs (if applicable)
**08–09**  Continuation count/status pairs (if applicable)
**0A–0B**  Continuation count/status pairs (if applicable)
**0C–0D**
    Continuation count/status pairs (if applicable)
**0E–0F**  Continuation count/status pairs (if applicable)
**10–11**  Continuation count/status pairs (if applicable)
**12–13**  Continuation count/status pairs (if applicable)
**14–15**  Continuation count/status pairs (if applicable)
**16–17**  Continuation count/status pairs (if applicable)
**18–19**  Continuation count/status pairs (if applicable)

**1A–1B** Continuation count/status pairs (if applicable)

**1C–1D**

Continuation count/status pairs (if applicable)

**1E–1F** Continuation count/status pairs (if applicable)

# SPT entry for SPT macro invoked (Part 1)

**Entry: SPT**
**VIT option:**
SSCP
**Event:** SPT macro invocation
**VIT processing module:**
ISTRACSC
**Control is returned to:**
ISTCPMSP

This trace record is written when partner LU mapping information in the generic resource coupling facility structure is created, deleted, or changed using the SPT macroinstruction.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---------|---|---|---|---|-----------------|---------|------------------|---------|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| SPT | I D | R T N C D | F U N C | F L A G S | GENERIC NAME | RETURN ADDRESS | REAL RESOURCE NAME | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"SPT"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code from the SPT macro

**X'00'** Request completed successfully. This is an existing USERVAR return code.

**X'01'** Request completed successfully and local data was returned. This is returned from SPT if SEARCH is CF&LOCAL only.

**X'04'** Resource specified could not be located. This is an existing USERVAR return code.

**X'51'** Request failed due to APPC restrictions. This is an existing USERVAR return code.

**X'52'** Request failed due to a parameter specification that was not valid. This is an existing USERVAR return code.

**X'53'** The function requested could not be completed because of a storage allocation failure. This is an existing USERVAR return code.

**X'54'** The real instance of user variable to be deleted could not be located. This is an existing USERVAR return code.

**X'55'** A conflict was found due to NETIDs.

**X'56'** A conflict was found due to CPNAMEs.

**X'57'** VTAM is halting.

**X'58'** ENDAFFINITY was issued for a session that was VTAM owned.

**X'59'** RSCTYPE value conflicted between USERVAR and GR.

**X'5A'** An attempt to change the generic name for a given application failed because SPTEs pertaining to the previous generic name still exist.

**X'60'** A session pair could not be found in an SPT entry.

**X'61'** An attempt to change the generic name for a given application failed because either (1) the existing generic name was never deleted, or (2) because SPTEs pertaining to the existing generic name still exist.

**X'62'** An attempt to repopulate the generic resource coupling facility structure has failed because the local data being used for repopulation is backlevel compared to data already in the structure.

**X'63'** An attempt to repopulate the generic resource coupling facility structure has succeeded but backlevel data has been identified in the structure and additional structure cleanup may be required.

**X'64'** An attempt to increment the session count for an SPTE failed because the name type of the SPTE did not match what was expected.

**X'65'** Request failed due to TSO mismatch.

**X'66'** An attempt to register a generic resource with the workload manager failed because a STOKEN that was not valid was used.

**X'A0'** A request made against the generic resources coupling facility structure will complete synchronously.

**X'A1'** An attempt to update information in the generic resources coupling facility structure failed because that data had changed since it was last read. The data should be reread and then modified again.

**X'A2'** An attempt to access the generic resources coupling facility structure failed for an unexpected reason.

**X'A3'** There is currently no connection to the generic resources coupling facility structure.

**X'A4'** The buffers provided for reading data from the generic resources coupling facility structure were insufficient for buffering all the data associated with the list entry being read. No data (adjunct or element) is returned.

**X'A5'** A read from the generic resources coupling facility structure failed because the requested data could not be found in that structure.

**X'A6'** Data could not be added to the generic resources coupling facility structure because there is insufficient storage in the generic resources coupling facility structure to hold it.

**06** Reason macro invoked:

**X'01'** Find an SPT entry.

**X'02'** Add a session pair to the SPT entry.

**X'03'** Decrement the session count for a session pair.

**X'04'** Increment the session count for a session pair.

**X'05'** End the affinity between the generic resource and its partner LU.

**X'06'** Update an affinity between the generic resource and its partner LU.

**X'07'** Delete a session pair in an SPT entry.

**X'08'** Free local SPT entry that was read from the coupling facility structure.

**X'09'** Create a local copy of an SPT entry.

**07** Flags:

**x... ....**
    Name type
    **B"0"** Real names only

> **B"1"** Generic names allowed
>
> **.1.. ....**
>> This SPT will persist until the application issues the
>> CHANGE OPTCD=ENDAFFIN macroinstruction. Refer to *z/OS*
>> *Communications Server: SNA Programming* for additional
>> information.
>
> **..x. ....**
>> Ownership indicator
>> **B"0"** VTAM-owned
>> **B"1"** Application-owned
>
> **...x ....**
>> Multinode persistent session indicator
>> **B"0"** SPT entry is not associated with a multinode persistent
>> session application program
>> **B"1"** SPT entry is associated with a multinode persistent session
>> application program
>
> **.... xx..**
>> Affinity creator.
>> **B"00"** The affinity was created during a previous request.
>> **B"01"** VTAM selected the instance of the generic resource.
>> **B"10"** The MVS WLM selected the instance of the generic
>> resource.
>> **B"11"** The Generic Resources Exit selected the instance of the
>> generic resource.
>
> **.... ..1.**
>> This affinity is for an LU 6.2 session that is not a user sync point or
>> a limited resource.
>
> **.... ...x**
>> 0

**08–0F** Generic resource name
**10–13** Address of the issuer of the SPT macro
**14–1B** Application program network name
**1C–1F** Request parameter header (RPH) address

# SPT2 entry for SPT macro invoked (Part 2)

**Entry: SPT2**
**VIT option:**
> SSCP

**Event:** SPT macro invocation
**VIT processing module:**
> ISTRACSC

This trace record is a continuation of the SPT entry.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|
| SPT2 | SIB COUNT | PCID | NETID OF PARTNER LU | PARTNER LU NAME |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"SPT2"
**04–07** Number of session information blocks (SIBs) that have been created for this SPT entry
**08–0F** PCID if associated with a session, otherwise 0
**10–17** Partner NETID
**18–1F** Partner resource name

## SRBD entry for SRB dispatch

**Entry: SRBD**
**VIT option:**
  PSS
**Event:** SRB dispatch
**VIT processing module:**
  ISTRACPS
**Control is returned to:**
  ISTAPCAD

This trace record provides information when PSS is dispatched under an SRB.

**Note:** For this trace record to be created, in addition to specifying the PSS option, you must specify SRB or BOTH on the PSS TRACE start option.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 | 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D | E F |
|---|---|---|---|---|---|---|---|---|---|
| SRBD | ID | 0 | PST ADDRESS | PST LIFO ASYNC DSP QUEUE | PST FIFO ASYNC DSP QUEUE | PST ASYNC N-DSP QUEUE (ANDSP) | TCB ADDRESS | F L G 1 | 0 |

**Byte (hex)**
  **Contents**
**00–03** Record ID: C"SRBD"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07** 0
**08–0B** PST address
**0C–0F** PST LIFO asynchronous dispatchable queue (PSTADSP)
**10–13** PST FIFO asynchronous dispatchable queue (PSTALIST)
**14–17** PST asynchronous nondispatchable queue (PSTANDSP)
**18–1B** TCB address (PSTTCBA)
**1C–1D**
  PST flags (PSTFLG1)
**1E–1F** 0

## SRBX entry for SRB exit

**Entry: SRBX**

**VIT option:**
> PSS

**Event:** SRB exit

**VIT processing module:**
> ISTRACPS

**Control is returned to:**
> ISTAPCSX, ISTAPCTX, or ISTAPCAD

This trace record provides information when PSS exits an SRB. For this trace record to be created, you must specify the PSS VIT option, and you must specify SRB or BOTH on the PSSTRACE start option.

| 0 0 0 0<br>0 1 2 3 | 0<br>4 | 0<br>5 | 0<br>6 7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1 1 1<br>0 1 2 3 | 1 1 1 1<br>4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|---|---|
| SRBX | I<br>D | C<br>O<br>D<br>E | 0 | PST<br>ADDRESS | PST<br>LIFO<br>ASYNC<br>DSP<br>QUEUE | PST<br>FIFO<br>ASYNC<br>DSP<br>QUEUE | NEW<br>ASCB<br>ADDRESS | OLD<br>ASCB<br>ADDRESS | RETURN<br>ADDRESS |

**Byte (hex)**
> **Contents**

| | |
|---|---|
| 00–03 | Record ID: C"SRBX" |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | Reason code for SRB exit |
| 06–07 | 0 |
| 08–0B | PST address |
| 0C–0F | PST LIFO asynchronous dispatchable queue (PSTADSP) |
| 10–13 | PST FIFO asynchronous dispatchable queue (PSTALIST) |
| 14–17 | New ASCB address |
| 18–1B | Old ASCB address |
| 1C–1F | Address of the module issuing the SRBX trace event |

# SRTx entry for SRTADD, SRTCHG, SRTDEL, SRTFIND macros

**Entry: SRTA, SRTC, SRTD, or SRTF**

**VIT option:**
> NRM

**Event:** SRTADD, SRTCHG, SRTDEL, SRTFIND macros

**VIT processing module:**
> ISTRACNR

**Control is returned to:**
> ISTNRCSA for SRTADD, ISTNRCSD for SRTCHG, SRTDEL, and SRTFIND

This trace record contains information about a SRTADD, SRTCHG, SRTDEL, or SRTFIND macroinstruction issued by a VTAM module.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SRTA SRTC SRTD SRTF | | | | ID | RTNCD | SRTUSELN | TYPE | HASH NAME (NAME OR NETWORK ADDRESS) | | | | | | | | RETURN ADDRESS | | | | SRT ENTRY ADDRESS | | | | NETWORK ID | | | | | | | |

**Byte (hex)**
**Contents**

**00–03** Record ID: C"SRTA" (SRTADD), C"SRTC"(SRTCHG), C"SRTD" (SRTDEL), or C"SRTF" (SRTFIND)

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code. Refer to the information about the SRT control block in *z/OS Communications Server: SNA Data Areas Volume 1*.

**06** 0 or SRTUSELN from the SRTE. This is a 4-byte field in the SRTE and will be set to X'FF' in the SRTx if SRTUSELN is greater than X'FF'. This field will be 0 if the return code is nonzero, or if the SRT type does not use SRTUSELN as a use count.

**07** Type of SRT entry. For the types, refer to the information about the SRT control block in *z/OS Communications Server: SNA Data Areas Volume 1*.

**08–0F** Network name or network address (padded on left with zeros) passed to the hashing algorithm

**10–13** Address of the issuer of the SRTADD, SRTCHG, SRTDEL, or SRTFIND macroinstruction

**14–17** Address of SRT entry for the resource

**18–1F** Network ID of the resource

# SRT2 entry for the SRTADD macro

**Entry: SRT2**
**VIT option:**
    NRM
**Event: SRTADD**
**VIT processing module:**
    ISTRACNR

This trace record is a continuation of the SRTADD entry. It is written after an SRTADD failure if the duplicate SRTE is contained in or points to an RDTE.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SRT2 | | | | SRTUSELN | FLAGS | RPRRENTRY | TYPE | RPRNAME | | | | | | | | SRTDATA OR 0 | | | | | | | | NETWORK ID | | | | | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"SRT2"

**04** SRTUSELN from the duplicate SRTE; this is a 4-byte field in the SRTE and will be set to X'FF' in the SRT2 if SRTUSELN is greater than X'FF'.

**05** The first bit is SRTSPECE; the other seven are not used.

**06** RPRENTRY of the duplicate RDTE

**07** SRTTYP from the duplicate SRTE

**X'00'** RDTE

**X'03'** Shadow

**X'09'** Alias name

**X'0A'** Network address

**X'FF'** SRTE

**08–0F** RPRNAME of the duplicate RDTE

**10–13** SRTDATA from the duplicate SRTE or 0

**14–17** Not used

**18–1F** Network ID of the duplicate RDTE. If RPRENTRY is RPRENTRC, it is RCDNETID; otherwise, it is ATCNETID.

# TGM entry for APPN TG management (Part 1)

**Entry: TGM**

**VIT option:**
 SSCP

**Event:** APPN TG management event

**VIT processing module:**
 ISTRACSC

**Control is returned to:**
 ISTATCTR

This record is generated when one of the following APPN TG management events has occurred:

- Add an ATGB to an ATGB chain.
- Remove an ATGB from an ATGB chain.
- Find an ATGB.
- Activate a TG number.
- Deactivate a TG number.
- Negotiate a TG number.

If TGFIND fails to locate an ATGB, a trace entry will still be recorded.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TGM | | | TYPE | ID | | NN | RC | 0 | | ADJCP ADDR | | | | DYNF | CONC | CPCT | TGFS | CPNAME ADJACENT NODE | | | | | | | | CALLER NAME | | | RETURN ADDR | | |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"TGM"

**03** Type field

**Code Meaning**

| | |
|---|---|
| **C** | Activate transmission group number (TGN) or APPN transmission group block (ATGB) |
| **D** | Deactivate TGN or ATGB |
| **A** | Add an ATGB |
| **R** | Remove an ATGB |
| **F** | Find an ATGB |
| **N** | Negotiate a TGN |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | APPN TG Management function values |

- Network node specification
  - 0: Not a network node
  - 1: Network node
- RU being processed
  - 0: Request Contact
  - 1: Contacted
- TG number coded on macroinstruction
  - 0: Not specified
  - 1: Specified
- Display CV47
- APPN connection pending, not reported to topology (LIMBO) specification
  - 0: No
  - 1: Yes
- Vector format
  - 0: Key-length format
  - 1: Length-type format
- Assigned chain specification
  - 0: Assigned not specified
  - 1: Assigned specified
- Unassigned chain specification
  - 0: Unassigned not specified
  - 1: Unassigned specified

| | |
|---|---|
| **06** | The calling module's return code. 0 if no return address is provided. |
| **07** | Unused available |
| **08–0B** | Adjacent control point (ADJCP) address |
| **0C** | Dynamic indicators received in XID3 exchange (ADJDYNFL) |
| **0D** | Number of active connections controlled (ADJCONCT), last digit |
| **0E** | Connections controlled by the ADJCP (ADJCPCT), last digit |
| **0F** | Transmission group number state map (ADJSTMP), indexed by TGN |
| **10–17** | CPNAME of the adjacent node |
| **18–1B** | Calling module name |
| **1C–1F** | Return address |

# TGM2 entry for APPN TG management (Part 2)

**Entry: TGM2**
**VIT option:**
 SSCP
**Event:** APPN TG management event (Part 2)
**VIT processing module:**
 ISTRACSC

This trace record is a continuation of the TGM trace record.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| TGM2 | PU OF ATGB OR RCV TGN | ATGB ADDRESS | T G N U / T O P R / D Y N F / P O S | PUPTR<br><br>ATGB TOP | NXTPT<br><br>ATGB TOP | PUPTR<br><br>ATGB NEXT | NXTPT<br><br>ATGB NEXT |

**Byte (hex)**
> **Contents**

**00–07** Overlay 1
  - 00-03 Record ID: C"TGM2"
  - 04-07 PU of current ATGB

  Overlay 2
  - 00-03 Record ID: C"TGM2"
  - 04-06: Unused, available
  - 07: Received TGN

**08–0B** ATGB address
**0C** ATGTGNUM
**0D** ATGTOPR
**0E** ATGDYNFL
**0F** ATGPOS
**10–13** ATGPUPTR: Top ATGB in the chain
**14–17** ATGNXTPT: Top ATGB in the chain
**18–1B** ATGPUPTR: Next ATGB in the chain
**1C–1F** ATGNXTPT: Next ATGB in the chain

# TGVC entry for Transmission Group Vector (Part 1)

**Entry:** TGVC
**VIT option:**
> SSCP

**Subtrace Type:**
> TGVC

**Event:** Request Route, Recompute Route, Request TG Vectors, or Cache Data message
**VIT processing module:**
> ISTITCTG

**Control is returned to:**
> Module invoking the INTRACE TYPE(TGVC) macro

This record is generated when the subtrace TGVC under SSCP trace option is active and a TRS Request Route, Recompute Route, Request TG Vectors, or SSL Cache Data message is requested.

**Note:** Because of the potentially large amounts of data contained in the TG Vectors, it is not recommended that this function be turned on, except for problem diagnosis. If this function is used, it should only be turned on for the time required to generate the necessary documentation and then turned off.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| TGVC | A S I D | 0 | T Y P E | C O N T I N U E | PCID | TG VECTOR DATA |

**Byte (hex)**
**Contents**
**00–03**  Record ID: C"TGVC"
**04**  ASID is the primary address space ID
**05**  0
**06**  Trace type
  • C"P"– PLU
  • C"S"– SLU
  • C"U"– Unknown
**07**  Continuation trace
  • C"F"– First set of entries
  • C"C"– Continuation set of entries
**08–0F**  PCID
**10–1F**  TG Vector data

# TGV2 entry for Transmission Group Vector (Part 2)

**Entry:  TGV2**
**VIT option:**
  SSCP
**Subtrace Type:**
  TGVC
**Event:**  Request Route, Recompute Route, Request TG Vectors, or Cache Data
  message
**VIT processing module:**
  ISTITCTG

This record is a continuation of the TGVC trace record.

**Note:** If the complete CV46/CV47 pair will not fit in the remaining TGV2
  record(s), another TGVC/TGV2 set will be generated.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| TGV2 | TG VECTOR DATA |

**Byte (hex)**
**Contents**
**00–03**  Record ID: C"TGV2"
**04–1F**  TG Vector Data

## TOKx entry for token collection

**Entry: TOKA or TOKD**
**VIT option:**
 CIA
**Event:** Addition or deletion to token collection
**VIT processing module:**
 ISTRACCI
**Control is returned to:**
 Module invoking the INTRACE macroinstruction that caused the record to
 be produced

The TOKA trace record is written when a token is added to one of the token
collections that are maintained by Registration Manager. The TOKD trace record is
written when a token is deleted from one of the token collections that are
maintained by Registration Manager.

| 0 0 | 0 0 | 0 | 0 | 0 0 | 0 0 | 0 0 | 0 0 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 | 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 | 6 7 | 8 9 | A B | C D | E F | 0 1 | 2 3 | 4 5 | 6 7 | 8 9 | A B | C D E F |
| TOKA TOKD | | ID | 0 | 0 | MODULE ID | | PTOKEN | | 0 | | OBJECT TOKEN | | CALLER ADDR | | RPH ADDR |

**Byte (hex)**
 **Contents**
**00–03** Record ID:
 • C"TOKA" for addition of token
 • C"TOKD" for deletion of token
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is
 greater than X'FF'.
**05** 0
**06** 0
**07–0B** Module eyecatcher
**0C–0F** PTOKEN (Provider = D)
**10–13** 0
**14–17** Token of object being added or deleted (filter object)
**18–1B** Address of caller
**1C–1F** Request parameter header (RPH) address.

## TOPN entry for node modified (Part 1)

**Entry: TOPN**
**VIT option:**
 SSCP
**Event:** Topology node entry modified
**VIT processing module:**
 ISTITCST
**Control is returned to:**
 Module invoking the INTRACE TYPE(TOPN) macro

This record is generated when a node entry in the topology database is created or
modified.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOPN | | | | I D | 0 | A C T I O N | 0 | CP NAME | | | | | | | | RETURN ADDRESS | | | | NODE HEADER POINTER | | | | 0 | | | F L A G S | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"TOPN"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Action performed on Topology Database:

**Code** **Meaning**
**C"A"** Add node record
**C"D"** Delete node record
**C"U"** Update node record

**07** 0

**08–0F** CP name of the node represented by this node record

**10–13** Return address of the module that changed the Topology Database

**14–17** Node header pointer

**18–1A** 0

**1B** Flag bytes as follows:

**Bit** **Meaning**
**11.. ....**
     Node type
     **B"00"** End node
     **B"01"** Network node
     **B"10"** Virtual routing node
**..1. ....**
     Indicate whether this node is nonnative
**...x xxxx**
     Not used

**1C–1F** Request parameter header (RPH) address

# TPN2 entry for node modified (Part 2)

**Entry:** **TPN2**
**VIT option:**
     SSCP
**Event:** Topology node entry modified
**VIT processing module:**
     ISTITCST

This record is a continuation of the TOPN trace record.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 | 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 | 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E | F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C | D E F |
|---|---|---|---|---|---|---|---|---|---|
| TPN2 | NODE RECORD POINTER | RESOURC SEQUENC NUMBER | V4580 FLAGS | R E A S O N | 0 | | NETWORK ID | T I M E | 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"TPN2"

**04–07** Node record pointer

**08–0B** Current resource sequence number for the node

**0C–0E** Node characteristics control vector flags (V4580_FLAGS)

| Bit | Meaning |
|---|---|
| **1... ....** | Node congested |
| **.1.. ....** | Intermediate routing resources depleted |
| **..1. ....** | Endpoint resources depleted |
| **...1 ....** | Garbage collection indicator |
| **.... 1...** | High-order bit of the 3-bit Locate Message Size supported by this node |
| **.... .1..** | Node is shutting down |
| **.... ..11** | Low-order bits of the 3-bit Locate Message Size supported by this node |

**0F** Garbage collection indicator

**10–13** 0

**14–1B** Network ID

**1C** Number of days left before this node is garbage collected

**1D–1F** 0

# TOPT entry for TG modified (Part 1)

**Entry: TOPT**

**VIT option:**
> SSCP

**Event:** Topology TG entry modified

**VIT processing module:**
> ISTITCST

**Control is returned to:**
> Module invoking the INTRACE TYPE(TOPT) macro.

This record is generated when a TG entry in the topology database is created or modified.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| TOPT | | | | I D | 0 | A C T I O N | T G N U M B E R | TG ORIGIN NODE CP NAME | | | | | | | | RETURN ADDRESS | | | | TG DESTRINATION NODE CP NAME | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"TOPT"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Action performed on Topology Database:

**Code** **Meaning**
**C"A"** Add TG record
**C"D"** Delete TG record
**C"U"** Update TG record

**07** TG number

**08–0F** CP name of the TG origin node

**10–13** Return address of the module that changed the Topology Database

**14–1B** CP name of the TG destination node

**1C–1F** Request parameter header (RPH) address

# TPT2 entry for TG modified (Part 2)

**Entry: TPT2**
**VIT option:**
SSCP
**Event:** Topology node entry modified
**VIT processing module:**
ISTITCST

This record is a continuation of the TOPN trace record.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 | 0 | 0 | 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 | 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C | D | E | F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C | D E F |
| TPT2 | TG RECORD POINTER | RESOURC SEQUENC NUMNBER | V 4 7 S T A T | V 4 6 8 0 F L G | F L A G S | R E A S O N | ORIGIN NODE RECORD POINTER | DEST NODE RECORD POINTER | REVERSE TG RECORD POINTER | T I M E | 0 |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"TPT2"

**04–07** TG record pointer

**08–0B** Current resource sequence number for the TG

**0C** TG characteristics status flags (V47STAT)

**0D** TG descriptor flags (V4680FLAGS)

**0E** TG information flags

| Bit | Meaning |
|---|---|
| **111. ....** | TG type |
| | **B"000"** Endpoint TG |
| | **B"001"** Intermediate routing TG |
| | **B"010"** Interchange TG |
| | **B"011"** Intersubnet TG |
| | **B"100"** Branch extender TG |
| **...1 ....** | Indicate whether HPR control flows supported |
| **.... xxxx** | Not used |

**0F** Garbage collection indicator

    **C"G"** This entry is generated by Garbage Collection processing.

    **C"M"** This entry is generated by Routing Tree Marking processing.

**10–13** Node record pointer of the TG origin node

**14–17** Node record pointer of the TG destination node

**18–1B** TG record pointer of the reverse TG

**1C** Number of days left before this TG is garbage collected

**1D–1F** 0

# TPT3 entry for TG modified (Part 3)

**Entry: TPT3**

**VIT option:**

    SSCP

**Event:** Topology node entry modified

**VIT processing module:**

    ISTITCST

**Control is returned to:**

    Module invoking the INTRACE TYPE(TOPT) macro.

This record is a continuation of the TPT2 trace record. This record will present only if the TG origin node has a different Network ID than the TG destination node.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TPT3 | | | | TG ORIGIN NODE NETWORK ID | | | | | | | | TG DESTINATION NODE NETWORK ID | | | | | | | | 0 | | | | 0 | | | | 0 | | | |

**Byte (hex)**
**Contents**
00–03  Record ID: C"TPT3"
04–0B  Network ID of the TG origin node
0C–13  Network ID of the TG destination node
14–1F  0

# TREx entry for routing tree Build or Update

**Entry:  TREM, TRED, TREI, or TREB**
**VIT option:**
     SSCP
**Event:  Routing tree Build or Update**
**VIT processing module:**
     ISTRACSC
**Control is returned to:**
     Module invoking the INTRACE macro that caused the record to be produced

A routing tree operations (TRE) VIT entry is created by topology and routing services (TRS) whenever one of the following events is encountered:

- TRS accesses or builds a routing tree to satisfy a request for an APPN route.
- An existing routing tree is marked for modification due to a topological change in the APPN network.
- TRS learns of the availability or loss of a central directory server or interchange node.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TREM TRED TREI TREB | | | | ID | MARK FLAGS | ADD OR DEL | TREE FLAGS | PATH WEIGHT OR RES SEQ NUMBER | | | | TREE HEADER POINTER OR TOPO DB RES POINTER | | | | RETURN ADDRESS | | | | ORIGIN TREE RECORD POINTER OR COS POINTER | | | | DEST TREE RECORD POINTER OR CALLING MODULE NAME | | | | TREE BUILD EXEC TIME | | | |

**Byte (hex)**

**00–03** Record ID:

> **C"TREM"**
>> Mark tree.
>
> **C"TRED"**
>> Add or change directory server.
>
> **C"TREI"**
>> Add or change interchange node.
>
> **C"TREB"**
>> Build tree.

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Tree marking flags

> **Bit** **Meaning**
>
> **x... ....**
>> 1=Operable resource state
>> 0=Inoperable resource state
>
> **.xxx ....**
>> Reason
>>
>> 000=Transmission group update
>> 001=Topology database update
>> 010=Garbage collection
>> 011=Overuse
>> 100=Modify topology
>> 110=Uncache tree
>> 111=Unreachable partner change
>
> **.... xxxx**
>> 0

**06** Indicates that a directory server or interchange node is added ("A") or deleted ("D")

**07** Tree building flags

> **Bit** **Meaning**
>
> **x... ....**
>> Indicates whether existing tree is used
>>
>> 1=Existing tree is used.
>> 0=New tree is built; existing tree is not available.
>
> **.x.. ....**
>> Indicates whether clean path is used.
>>
>> 1=Clean path is used.
>> 0=Clean path is not available.
>
> **..x. ....**
>> Indicates number of destinations
>>
>> 1=Multiple
>> 0=Single

**08–0B** Indicates the entry type.

> **Entry type**
>> **Meaning**
>
> **B** Path weight
> **M** Resource sequence number

> **Note:** If entry type is TREB and the destination tree record pointer indicates multiple destinations, the path weight field is not valid.

**0C–0F** Tree header pointer or topology database resource pointer, depending on entry type:

| | **Meaning** |
|---|---|
| **B** | Tree header pointer |
| **M** | Topology database resource pointer |

**10–13** Return address of caller

**14–17** Origin tree record pointer or CoS pointer
- Origin tree, if entry type is B
- CoS pointer, if tree marking flags are ('110')

**18–1B** Destination tree record pointer or calling module
- Destination tree record pointer, if entry type is B
- Last four characters of calling module name, if tree marking flags are ('110')

**1C–1F** Tree building execution time (microseconds), if entry type is B

# TRMR entry for mark record (Part 1)

**Entry: TRMR**
**VIT option:**
 SSCP
**Event:** Routing tree marked for future update
**VIT processing module:**
 ISTITCAB
**Control is returned to:**
 ISTTRRUM

This record is generated when TRS marks a tree record due to changes on the network topology.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| TRMR | A S I D | 0 | I N S T A N C E | S T A T U S | MARKED TREE RECORD POINTER | TREE HEADER POINTER | MARKER NODE RECORD POINTER | NODE HEADER POINTER | ROOT TREE RECORD POINTER | RPH ADDRESS |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"TRMR"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Trace entry instance
**07** New status of the tree record being marked

 **Code** **Meaning**
 **C"C"** Tree record is clean, no changes on this tree record.

**C"I"** Tree record is incomplete, a change was detected on a descendant of this tree record but the tree has not been updated.

**C"D"** Tree record is dirty, a change was detected on this tree record but the tree has not been updated.

**08–0B** Tree record pointer
**0C–0F** Tree header pointer
**10–13** Node record pointer
**14–17** Node header pointer
**18–1B** Tree record pointer for the root of the routing tree
**1C–1F** Request parameter header (RPH) address

## TRM2 entry for mark record (Part 2)

**Entry: TRM2**
**VIT option:**
      SSCP
**Event:** Routing tree marked for future update
**VIT processing module:**
      ISTITCAB

This record is a continuation of the TRM2 trace record.

| 0 0 0 0 0 0 0 0<br>0 1 2 3 4 5 6 7 | 0 0 0 0 0 0 0 0<br>8 9 A B C D E F | 1 1 1 1 1 1 1 1<br>0 1 2 3 4 5 6 7 | 1 1 1 1 1 1 1 1<br>8 9 A B C D E F |
|---|---|---|---|
| TRM2 · A C T I O N · T Y P E · 0 | ROOT<br><br>TREE<br><br>CP NAME | MARKER<br><br>TREE<br><br>RECORD<br><br>CP NAME | COS<br><br>NAME |

**Byte (hex)**
      **Contents**
**00–03** Record ID: C"TRM2"
**04** Action on the topology entry that caused the tree marking

      **Code** **Meaning**
      **C"I"** Incomplete
      **C"U"** Update
      **C"A"** Add
      **C"D"** Delete
**05** Type of topology entry

      **Code** **Meaning**
      **C"T"** TG
      **C"N"** Node
**06–07** 0
**08–0F** CP name of the node on tree root
**10–17** CP name of the tree record being marked
**18–1F** CoS name

## TRM3 entry for mark record (Part 3)

**Entry: TRM3**

**VIT option:**
> SSCP

**Event:** Routing tree marked for future update

**VIT processing module:**
> ISTITCAB

This record is a continuation of the TRM2 trace record. TRM3 is generated only if any of the CP names in the TRM2 record has a different network ID than the local node.

| 0 0 0 0<br>0 1 2 3 | 0 0 0 0<br>4 5 6 7 | 0 0 0 0 0 0 0 0<br>8 9 A B C D E F | 1 1 1 1 1 1 1 1<br>0 1 2 3 4 5 6 7 | 1 1 1 1 1 1 1 1<br>8 9 A B C D E F |
|---|---|---|---|---|
| TRM3 | 0 | ROOT<br><br>TREE<br><br>RECORD<br><br>NETWORK ID | MARKER<br><br>TREE<br><br>RECORD<br><br>NETWORK ID | 0 |

**Byte (hex)**
> **Contents**

00–03    Record ID: C"TRM3"

04–07    0

08–0F    Network ID of the node on tree root

10–17    Network ID of the tree record being marked

18–1F    0

# TRNM entry for translate message (Part 1)

**Entry:**  **TRNM**

**VIT option:**
> MSG

**Event:**  Translate message error (Part 1)

**VIT processing module:**
> ISTRACOT

**Control is returned to:**
> ISTUSCMS or IKTMSIFR

This trace record is generated when a return code greater than the value 4 is received from the TRANMSG macro. When the return code and reason code indicate that the MVS message service is not active, no TRNM trace record is generated. A single message is translated each time the TRANMSG macro is invoked.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| TRNM | I D | LANG CODE | MESSAGE I/O BLOCK ADDRESS | MESSAGE I/O BLOCK LENGTH | RETURN CODE | REASON CODE | MPB ADDRESS | MTB ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"TRNM"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07** 3-character language code or blanks if language is not supported

| Hex Value | Language Code | Language Name |
|---|---|---|
| X'02' | ARA | Arabic |
| X'03' | CHT | Traditional Chinese |
| X'04' | CHS | Simplified Chinese |
| X'05' | DAN | Danish |
| X'06' | DEU | German |
| X'07' | DES | Swiss German |
| X'08' | ELL | Greek |
| X'09' | ENG | UK English |
| X'00' | | US English (default) |
| X'01' | ENU | US English (specified) |
| X'0A' | ESP | Spanish |
| X'0B' | FIN | Finnish |
| X'0C' | FRA | French |
| X'0D' | FRB | Belgian French |
| X'0E' | FRC | Canadian French |
| X'0F' | FRS | Swiss French |
| X'10' | HEB | Hebrew |
| X'12' | ISL | Icelandic |
| X'13' | ITA | Italian |
| X'14' | ITS | Swiss Italian |
| X'11' | JPN | Japanese |
| X'15' | KOR | Korean |
| X'16' | NLD | Dutch |
| X'17' | NLB | Belgian Dutch |
| X'18' | NOR | Norwegian |
| X'19' | PTG | Portuguese |
| X'1A' | PTB | Brazil Portuguese |
| X'1B' | RMS | Rhaeto-Romanic |
| X'1C' | RUS | Russian |
| X'1D' | SVE | Swedish |
| X'1E' | THA | Thai |
| X'1F' | TRK | Turkish |
| X'3F' | | Unknown language code |

**08–0B** Message input/output block address
**0C–0F** Message input/output block length
**10–13** Return code
**14–17** Reason code

18–1B   Input message parameter block address (MIOINPTP). If a message text block or text is supplied instead, 18–1B is the address of the text block or text.

1C–1F   Output message text block address (MIOBUFFP)

## TRN2 entry for translate message (Part 2)

**Entry:  TRN2**
**VIT option:**
>   MSG

**Event:  Translate message error (Part 2)**
**VIT processing module:**
>   ISTRACOT

This trace record is a continuation of the TRNM entry. It is generated when a TRNM entry is produced and the input to the TRANMSG macro is the address of a message parameter block.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
|---|---|---|---|---|
| TRN2 | IST, IKT, OR USS MESSAGE NUMBER | FIRST MESSAGE VARIABLE OR 0 | SECOND MESSAGE VARIABLE OR 0 | THIRD MESSAGE VARIABLE OR 0 |

**Byte (hex)**
>   **Contents**

00–03   Record ID: C"TRN2"

04–07   Last four digits of the message ID (for example, 0200 for message IKT0200I, or SG02 for USSMSG02)

08–0F   First eight bytes of the first message variable, if present; otherwise, 0

10–17   First eight bytes of the second message variable, if present; otherwise, 0

18–1F   First eight bytes of the third message variable, if present; otherwise, 0

## TRRT entry for subtrace tree (Part 1)

**Entry:  TRRT**
**VIT option:**
>   SSCP

**Subtrace type:**
>   TREE

**Event:  APPN route computation**
**VIT processing module:**
>   ISTITCAB

**Control is returned to:**
>   Module invoking the INTRACE TYPE(TRRT) macro.

This record is generated when:

- Subtrace TREE under SSCP trace option is active
- TRS accesses a routing tree to satisfy a request for an APPN route

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRRT | | | | A S I D | 0 | I N S T A N C E | S T A T U S | CURRENT NODE RECORD POINTER | | | | CHILD NODE RECORD POINTER | | | | SIBLING NODE RECORD POINTER | | | | PARENT NODE RECORD POINTER | | | | T G W E I G H T | | N O D E W E I G H T | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"TRRT"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Trace entry instance

**07** Status on tree record for current node:

**Code** **Meaning**
C"C" Tree record is clean, no changes on this tree record.
C"I" Tree record is incomplete, a change was detected on a descendant of this tree record but the tree has not been updated.
C"D" Tree record is dirty, a change was detected on this tree record but the tree has not been updated.

**08–0B** Node record pointer of the node described by the current tree record

**0C–0F** Node record pointer of the node described by the child tree record

**10–13** Node record pointer of the node described by the sibling tree record

**14–17** Node record pointer of the node described by the parent tree record

**18–19** Weight of the TG from parent node to this node

**1A–1B** Weight of the current node

**1C–1F** Request parameter header (RPH) address

**Note:** An APPN route is comprised of TGs and network nodes. To compute the optimal route, TRS assigns a weight to each TG and network node by comparing the resource's characteristics against the characteristics required to set up the route. By adding up all weights, a route weight can be calculated and the optimal route with the least weight can be chosen.

## TRR2 entry for subtrace tree (Part 2)

**Entry: TRR2**
**VIT option:**
SSCP
**Subtrace type:**
TREE
**Event:** APPN route computation
**VIT processing module:**
ISTITCAB

This record is a continuation of the TRRT trace record.

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| TRR2 | T O T A L | W E I G H T | I N S T A N C E | 0 | CURRENT NODE CP NAME | CHILD NODE CP NAME | SIBLING NODE CP NAME |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–03 | Record ID: C"TRR2" |
| 04–05 | Total path weight from the root of routing tree to the current node |
| 06 | Trace entry instance |
| 07 | 0 |
| 08–0F | CP name of the current node |
| 10–17 | CP name of the child node |
| 18–1F | CP name of the sibling node |

# TRR3 entry for subtrace tree (Part 3)

**Entry: TRR3**
**VIT option:**
> SSCP

**Subtrace type:**
> TREE

**Event:** APPN route computation
**VIT processing module:**
> ISTITCAB

This record is a continuation of the TRR2 trace record. TRR3 is generated only if any CP names in the TRR2 record have a different network ID than the local node.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| TRR3 | 0 | CURRENT NODE NETWORK ID | CHILD NODE NETWORK ID | SIBLING NODE NETWORK ID |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–03 | Record ID: C"TRR3" |
| 04–07 | 0 |
| 08–0F | Current node network ID |

**10–17**  Parent node network ID
**18–1F**  Sibling node network ID

# TRR4 entry for subtrace tree (Part 4)

**Entry:  TRR4**
**VIT option:**
>  SSCP
**Subtrace type:**
>  TREE
**Event:**  APPN route computation
**VIT processing module:**
>  ISTITCAB

This record is generated if the current routing tree hop (tree record) described by the TRRR and TRR2 pair contains any TGs. If the current hop contains multiple equal-weights TGs, TRR4 is generated for each TG on the hop.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRR4 | | | | 0 | | INSTANCE | 0 | DESTINATION NODE CP NAME | | | | | | | | RESOURCE SEQUENCE NUMBER | | | | RSCV COUNTER | | RESOURCE USAGE | | TG NUM | V4680FLG | V47STAT | 0 | TG RECORD POINTER | | | |

**Byte (hex)**
>  **Contents**
**00–03**  Record ID: C"TRR4"
**04–05**  0
**06**  Trace entry instance
**07**  0
**08–0F**  CP name of the TG destination node
**10–13**  Resource sequence number
**14–15**  RSCV counter
**16–17**  Resource usage counter
**18**  TG number
**19**  TG descriptor flags
**1A**  TG status flags
**1B**  0
**1C–1F**  TG record pointer

# TRR5 entry for subtrace tree (Part 5)

**Entry:  TRR5**
**VIT option:**
>  SSCP
**Subtrace type:**
>  TREE
**Event:**  APPN route computation

**VIT processing module:**
　　　ISTITCAB

This record is a continuation of the TRR4 trace record. TRR5 is generated only if the destination CP name in the TRR4 record has a different network ID than the local node.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 |
|---------|---------|-----------------|-----------------|-----------------|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B C D E F |
| TRR5 | 0 | DESTINATION NODE NETWORK ID | 0 | 0 |

**Byte (hex)**
　　　**Contents**
**00–03**　Record ID: C"TRR5"
**04–07**　0
**08–0F**　Destination node network ID
**10–1F**　0

**Notes:**

1. The group TRRT, TRR2, TRR3, TRR4, and TRR5 describes a single hop (represented by a tree record) on a routing tree. If the hop contains multiple equal-weight TGs, multiple TRR4 trace records will be generated (one for each TG on that hop).

2. When a routing tree is accessed to satisfy a route request and the TREE subtrace is active, the whole routing tree will be traced. Each hop (tree record) on the tree will generate the group TRRT, TRR2, TRR3, TRR4, and TRR5. The processing module ISITCAB is called when the INTRACE TYPE (TRRT) is issued. ISTITCAB will traverse the routing tree to trace every tree record on the tree.

3. After every tree record on the tree has been traced, control will be returned to the module that invoked the INTRACE macro.

# TSNS entry to trace sense codes

**Entry: TSNS**
**VIT option:**
　　　PIU
**Event:** TSNS record
**VIT processing module:**
　　　ISTRACOT
**Control is returned to:**
　　　Module invoking the INTRACE macro that caused the record to be produced.

This trace record allows a given TSC module to trace a sense code at the time it is generated. The TSNS event is treated as an exception condition and is always traced, whether or not the PIU option is in effect.

| 0 0 | 0 0 | 0 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 | 2 3 | 4 5 | 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
|---|---|---|---|---|---|---|---|---|
| TSNS | | I D | 0 | I N S T A N C E | 0 | RETURN ADDR | CB ADDR | SENSE CODE | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"TSNS"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–06** 0

**07** Instance identifier in invoking module

**08–0F** 0

**10–13** Address of issuer of the TSNS macro

**14–17** Control block address or 0

**18–1B** Sense code

**1C–1F** Request parameter header (RPH) address

# UE entry for USER exit (DFASY or RESP)

**Entry: UE**
**VIT option:**
     API
**Event: User exit (DFASY or RESP)**
**VIT processing module:**
     ISTRACAP
**Control is returned to:**
     ISTAICUE

This trace record identifies the exit dispatched and data passed back to the user application program.

If this entry is associated with an event failure (that is, the return code is nonzero), this entry will be generated whether or not the API option is in effect. It is treated as an exception condition and is always traced, regardless of the VIT options specified, if the VIT is active.

For DFASY and RESP exits (type codes = 07 and 08), the user exit trace record has the following format:

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UE | | CODE | EXTDS | ID | RTNCD | FDB2 | FDB3 | RPL ADDRESS | | | | EXIT ADDRESS | | | | RPL AREA | | | | RPL RLEN | | | | CID OR 0 | | | | RPL FDBK2 | | | |

**Byte (hex)**

**Contents**

**00–01** Record ID: C"UE"

**02** Exit type code (hex 7=DFASY, hex 8=RESP)

**03** Exit definition (RPLEXTDS)

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Return code (RPLRTNCD) (For more information, refer to *z/OS Communications Server: SNA Programming*)

**06** Feedback code (RPLFDB2) (For more information, refer to *z/OS Communications Server: SNA Programming*)

**07** Feedback code (RPLFDB3) (For more information, refer to *z/OS Communications Server: SNA Programming*)

**08–0B** RPL address

**0C–0F** Exit address

**10–13** RPL data area pointer (RPLAREA) or 0

**14–17** Record length (RPLRLEN)

**18–1B** CID from NIB or RPL (RPLARG) or 0

**1C–1F** Second RPL feedback area, sense code (RPLFDBK2)

---

# UE1 entry for USER exit (not DFASY or RESP, Part 1)

**Entry: UE1**

**VIT option:**
API

**Event:** User exit (not DFASY or RESP, Part 1)

**VIT processing module:**
ISTRACAP

**Control is returned to:**
ISTAICUE

The user exit trace record (not DFASY or RESP, Part 1) has the following format:

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UE1 | | | | ID | CODE | 0 | | EXIT-TYPE DEPENDENT INFORMATION | | | | | | | | PRIMARY (APPL) LU NAME | | | | | | | | SECONDARY LU NAME OR 0 | | | | | | | |

**Byte (hex)**
**Contents**
**00–03**  Record ID: C"UE1"
**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**  Exit type code (see Table 7)
**06–07**  0
**08–0F**  Exit-type-dependent information (see Table 7)
**10–17**  Primary (application program) logical unit name
**18–1F**  Secondary logical unit name, or 0, if this is a TPEND exit

*Table 7. Exit-type-dependent information for a UE1 entry*

| Exit type code | Exit type | Byte (hex) | Contents |
|---|---|---|---|
| 05 | SCIP (BIND) | 08–09 | RPL control flags (RPLCNTDC, RPLCNTSC) |
| | | 0A–0F | Session parameters |
| 05 | SCIP (STSN) | 08–09 | RPL control flags (RPLCNTDC, RPLCNTSC) |
| | | 0A–0B | RPLOBSQV |
| | | 0C–0D | RPLIBSQV |
| | | 0E | RPLOBSQ |
| | | 0F | RPLIBSQ |
| 05 | SCIP (other) | 08–09 | RPL control flags (RPLCNTDC, RPLCNTSC) |
| | | 0C–0F | CID |
| 06 | LOGON | 08–0B | Logon data length |
| | | 0C–0F | CID |
| 09 | LOSTERM | 08–0B | Reason code |
| | | 0C–0F | CID |
| 0A | RELREQ | 08–0F | Unused |
| 0D | TPEND | 08–0B | Reason code |
| | | 0C–0F | Unused |
| 0E | NSEXIT | 08–0F | NS RU bytes 00–07 |

## UE2 entry for USER exit (Not DFASY or RESP, part 2)

**Entry:  UE2**
**VIT option:**
  API
**Event:**  User exit (not DFASY or RESP, Part 2)
**VIT processing module:**
  ISTRACAP

This trace record is a continuation of the UE1 entry.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| UE2 | | | | EXIT ADDRESS | | | | 0 | | | | | | | | | | | | | | | | NETID OR 0 | | | | | | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–03 | Record ID: C"UE2" |
| 04–07 | Exit address |
| 08–17 | 0 |
| 18–1F | Network identifier, or 0 if this is a TPEND exit |

# ULKA entry for release all locks

**Entry: ULKA**
**VIT option:**
　　LOCK
**Event:** Release all locks
**VIT processing module:**
　　ISTRACLK
**Control is returned to:**
　　ISTAPC39

This trace record contains information about a routine releasing all locks it currently holds.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| ULKA | | | | ID | 0 | 0 | | PST ADDRESS | | | | CRA LOCK ACCOUNT WORD | | | | RETURN ADDRESS | | | | 0 | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–03 | Record ID: C"ULKA" |
| 04 | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| 05 | 0 |
| 06–07 | 0 |
| 08–0B | PST address |
| 0C–0F | CRA lock account word (CRALKACT) |
| 10–13 | Address of the issuer of the TPUNLOCK macro |
| 14–1B | 0 |
| 1C–1F | Request parameter header (RPH) address |

# UNLK entry for release a lock

**Entry: UNLK**

**VIT option:**
 LOCK
**Event:** Release a lock
**VIT processing module:**
 ISTRACLK
**Control is returned to:**
 ISTAPC33

This trace record contains information for a routine releasing a lock. It is the complement of a LKEX or LKSH entry.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UNLK | | | | ID | 0 | LOCK LEVEL | | LOCK ADDRESS | | | | CRA LOCK ACCOUNT WORD | | | | RETURN ADDRESS | | | | LOCKWORD | | | | | | | L ID | RPH ADDRESS | | | |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"UNLK"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06–07** Level of lock to be released. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for a listing of VTAM locks.
**08–0B** Address of lock to be released
**0C–0F** CRA lock account word (CRALKACT)
**10–13** Address of the issuer of the TPUNLOCK macro
**14–1A** Lockword pointed to by address in bytes 08–0B
**1B** The LOCK ID field. Refer to *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* for a list of VTAM locks.
**1C–1F** Request parameter header (RPH) address

# UP entry for user application program post

**Entry:** UP
**VIT option:**
 API
**Event:** User application program post
**VIT processing module:**
 ISTRACAP
**Control is returned to:**
 ISTAICPT or ISTAPCSX

This trace record provides the RPL information that VTAM passes to the application program. The RPL information and the API TPIO request record (AIn or IOn) show VTAM activity for a given API request.

Refer to *z/OS Communications Server: SNA Programming* for a description of the RPL return codes.

If this entry is associated with an event failure (that is, the return code is nonzero), this entry will be generated whether or not the API option is in effect. It is treated as an exception condition, and is always traced if the VIT is active, regardless of the VIT options specified.

| 00 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UP | R E Q T Y | E X T D S | I D | R T N C D | F D B 2 | F D B 3 | RPL ADDRESS | EXIT OR ECB ADDRESS | RPL AREA | RPL RLEN | CID OR 0 | RPL FDBK2 |

**Byte (hex)**
>**Contents**
00–01    Record ID: C"UP"
02    RPL request type (RPLREQ)
>**X'15'**    SETLOGON
>**X'16'**    SIMLOGON
>**X'17'**    OPNDST
>**X'19'**    CHANGE
>**X'1A'**    INQUIRE
>**X'1B'**    INTRPRET
>**X'1F'**    CLSDST
>**X'22'**    SEND
>**X'23'**    RECEIVE
>**X'24'**    RESETSR
>**X'25'**    SESSIONC
>**X'27'**    SENDCMD
>**X'28'**    RCVCMD
>**X'29'**    REQSESS
>**X'2A'**    OPNSEC
>**X'2C'**    TERMSESS
03    Exit definition (RPLEXTDS)
04    ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05    Return code (RPLRTNCD) (Refer to *z/OS Communications Server: SNA Programming*.)
06    Feedback code (RPLFDB2) (Refer to *z/OS Communications Server: SNA Programming*.)
07    Feedback data flag (RPLFDB3) (Refer to *z/OS Communications Server: SNA Programming* under INQUIRE macro with OPTCD=APPSTAT.)
08–0B    RPL address
0C–0F    Exit address or ECB address
10–13    RPL data area pointer (RPLAREA)
14–17    Record length (RPLRLEN)
18–1B    CID from NIB or from RPL (RPLARG), or 0
1C–1F    Second RPL feedback area — Sense code (RPLFDBK2)

# USI or USO entry for APPC commands

**Entry:  USI or USO**
**VIT option:**
>APPC

**Event:** APPCCMD
**VIT processing module:**
    ISTRACAC
**Control is returned to:**
    Calling module

The USI trace record contains what RPLAREA points to when the following APPC commands are issued:

**APPCCMD**
    Data

**CONTROL=ALLOC,QUALIFY=\***
    FMH-5

**CONTROL=SENDFMH5**
    FMH-5

**CONTROL=OPRCNTL,QUALIFY=ACTSESS**
    Session parameters

**CONTROL=OPRCNTL,QUALIFY=CNOS**
    CNOS structure

**CONTROL=OPRCNTL,QUALIFY=DEFINE**
    DEFINE/DISPLAY structure

**Note:** * indicates all the possible values of QUALIFY when CONTROL equals ALLOC.

The USO trace record contains what RPLAREA points to when the following APPC commands complete:

**APPCCMD**
    Data

**CONTROL=OPRCNTL, QUALIFY=CNOS**
    CNOS structure

**CONTROL=OPRCNTL, QUALIFY=DISPLAY**
    DEFINE/DISPLAY structure

**CONTROL=OPRCNTL, QUALIFY=RESTORE**
    RESTORE structure

**CONTROL=RCVFMH5**
    FMH-5

A maximum of 212 bytes of user data will be traced. The FMH-5 will only be traced up to but not including the security access parameters.

**Note:** If the address of the CNOS session limits structure is not specified on the AREA keyword of the APPCCMD, the USI and USO records are not cut for the APPCCMD.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| USI OR USO | | | | ID | CNTRL | QUAL | 0 | RPL ADDRESS | | | | USER DATA | | | | | | | | | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

  **Contents**

**00–03** Record ID: C"USI" or C"USO"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** APPCCMD CONTROL operand value:

  **X'10'** ALLOC

  **X'12'** SENDFMH5

  **X'40'** OPRCNTL

  **X'60'** RCVFMH5

**06** APPCCMD QUALIFY operand value:

  **X'06'** CNOS

  **X'0C'** DEFINE

  **X'0D'** DISPLAY

  **X'12'** ACTSESS

  **X'14'** ALLOCD

  **X'15'** IMMED

  **X'16'** CONWIN

  **X'1B'** RESTORE

  **X'1E'** CONVGRP

  **X'1F'** WHENFREE

**07** 0

**08–0B** RPL address

**0C–1B** User data

**1C–1F** Request parameter header (RPH) address

# USI2 or USO2 entry for APPC commands

**Entry:** USI2 or USO2

**VIT option:**

  APPC

**Event:** APPCCMD

**VIT processing module:**

  ISTRACAC

This trace record is a continuation of the USI or USO entries. A maximum of 212 bytes of data is traced.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| USI2 OR USO2 | | | | USER DATA | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Byte (hex)**

> **Contents**

00–03  Record ID: C"USI2" or C"USO2"

04–1F  User data

# UVO or UVI entry for vector information

**Entry:  UVO or UVI**

**VIT option:**
> APPC

**Event:  VTRINA and VTRINL, or VTROUTA and VTROUTL specified on an APPCCMD macroinstruction**

**VIT processing module:**
> ISTRACAC

**Control is returned to:**
> APPCVL

This trace record contains vector information for APPCCMD macroinstructions that specify a vector area. UVI traces any input vectors (application-to-VTAM) when an APPCCMD macroinstruction is issued. UVO traces any output vectors when an APPCCMD macroinstruction is completed. The user data field of the USI and USO entries (0C–1B) is used to trace the total vector length (including the length of the length field) and vectors up to a maximum of eight trace records. The UVI vectors are traced from the application's storage. The UVO vectors are traced from VTAM's storage.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UVO OR UVI | | | | ID | CNTRL | QUAL | 0 | RPL ADDRESS | | | | VECTOR DATA | | | | | | | | | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

> **Contents**

00–03  Record ID:
- C"UVI" for input vector tracing
- C"UVO" for output vector tracing

04  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05  APPCCMD CONTROL operand value:

> **X'10'**  ALLOC

> **X'11'**  PREALLOC

|        |         |          |
|--------|---------|----------|
|        | **X'12'** | SENDFMH5 |
|        | **X'20'** | RESETRCV |
|        | **X'30'** | DEALLOC  |
|        | **X'31'** | DEALLOCQ |
|        | **X'40'** | OPRCNTL  |
|        | **X'50'** | PREPRCV  |
|        | **X'60'** | RCVFMH5  |
|        | **X'70'** | RECEIVE  |
|        | **X'71'** | RCVEXPD  |
|        | **X'80'** | REJECT   |
|        | **X'90'** | SEND     |
|        | **X'91'** | SENDEXPD |
|        | **X'92'** | SENDRCV  |
|        | **X'A0'** | SETSESS  |
|        | **X'B0'** | TESTSTAT |
| **06** | APPCCMD QUALIFY operand value: | |
|        | **X'00'** | NULL     |
|        | **X'01'** | ABNDPROG |
|        | **X'02'** | ABNDSERV |
|        | **X'03'** | ABNDTIME |
|        | **X'04'** | ABNDUSER |
|        | **X'05'** | ANY      |
|        | **X'06'** | CNOS     |
|        | **X'07'** | CONFIRM  |
|        | **X'08'** | CONFRMD  |
|        | **X'09'** | DATA     |
|        | **X'0A'** | DATACON  |
|        | **X'0B'** | DATAFLU  |
|        | **X'0C'** | DEFINE   |
|        | **X'0D'** | DISPLAY  |
|        | **X'0E'** | ERROR    |
|        | **X'0F'** | FLUSH    |
|        | **X'10'** | RQSEND   |
|        | **X'11'** | SPEC     |
|        | **X'12'** | ACTSESS  |
|        | **X'13'** | DACTSESS |
|        | **X'14'** | ALLOCD   |
|        | **X'15'** | IMMED    |
|        | **X'16'** | CONWIN   |
|        | **X'17'** | SESSION  |
|        | **X'18'** | CONV     |
|        | **X'19'** | SUSPEND  |
|        | **X'1A'** | RESUME   |
|        | **X'1B'** | RESTORE  |
|        | **X'1C'** | SYNCBEG  |
|        | **X'1D'** | SYNCEND  |
|        | **X'1E'** | CONVGRP  |
|        | **X'1F'** | WHENFREE |
|        | **X'20'** | IANY     |
|        | **X'21'** | ISPEC    |
|        | **X'22'** | ALL      |
|        | **X'23'** | IALL     |
|        | **X'24'** | QUEUE    |
|        | **X'25'** | DATAQUE  |
| **07** | 0       |          |
| **08–0B** | RPL address | |

**0C–1B** Contents of the RPL6VAIA for UVO trace records. Contents of the RPL6VAOA for UVI trace records. No more than eight trace records of vector data is traced.

**1C–1F** Request parameter header (RPH) address

## UVO2 or UVI2 entry for vector information

**Entry: UVO2 or UVI2**

**VIT option:**
    APPC

**Event:** VTRINA and VTRINL, or VTROUTA and VTROUTL specified on an APPCMD macroinstruction

**VIT processing module:**
    ISTRACAC

This trace record is a continuation of the UVO entry.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---------|-------------------------|---------------------------------|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| UVO2 OR UVI2 | VECTOR DATA | |

**Byte (hex)**

    **Contents**

**00–03** Record ID:
- C"UVI2" for continuation of UVI
- C"UVO2" for continuation of UVO

**04–1F** Vector data:
- Contents of the RPL6VAOA for UVI trace records.
- Contents of the RPL6VAIA for UVO trace records.

## VCC1 entry for CNCB queued (Part 1)

**Entry: VCC1**

**VIT option:**
    VCNS

**Event:** CNCB queued (Part 1)

**VIT processing module:**
    ISTRACNS

The VCC1 trace record is written when a CNS protocol control block (CNCB) is queued between CNS and LSA components. The VCC2 trace record displays data contained in the CNCB.

| 0 0 0 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B C D E F |
| VCC1 | I D | C B I D | S T A T E | C C O D E | MAJOR CONTROL BLOCK ADDRESS | CNCB ADDRESS | REQSTING MODULE ABBREV NAME | 12 BYTES OF CNCB DATA |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"VCC1"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Control block identifier

 **X'0B'** ISTVLNCB

 **X'7B'** ISTRIB

**06** Process state of the control block processing the dequeued work element:

 LSNCB states when CBID=X'0B' (ISTVLNCB)

 **X'00'** Reset

 **X'01'** Pending active

 **X'02'** Active without application logged on to line

 **X'03'** Active with application logged on to line

 **X'04'** Pending inactive phase 1

 **X'05'** Pending inactive phase 2

 **X'06'** Pending inactive phase 3

 **X'07'** Blocked

 States when CBID=X'7B' (ISTRIB)

 **X'00'** Idle

 **X'01'** Logon in progress

 **X'02'** Active

 **X'03'** Logoff in progress

 **X'04'** DACTLINK in progress

 **X'05'** Forced termination in progress

**07** CNCB primitive code

 **X'01'** CNS_INQUIRE_REQUEST

 **X'02'** CNS_INQUIRE_CONFIRM

 **X'03'** CNS_LOGON_REQUEST

 **X'04'** CNS_LOGON_CONFIRM

 **X'05'** CNS_LOGOFF_REQUEST

 **X'06'** CNS_LOGOFF_CONFIRM

 **X'07'** CNS_LOGOFF_RESPONSE

 **X'08'** CNS_LOGOFF_INDICATION

 **X'09'** CNS_CLOSEACB_REQUEST

 **X'0A'** CNS_CLOSEACB_CONFIRM

 **X'0B'** CNS_CLOSEACB_COMPLETION_INDICATION

**08–0B** Address of the major control block

**0C–0F** CNCB address

**10–13** Requesting module abbreviated name

**14–1F** First 12 bytes of CNCB data

# VCC2 entry for CNCB queued (Part 2)

**Entry: VCC2**

**VIT option:**
 VCNS
**Event:** CNCB queued (Part 2)
**VIT processing module:**
 ISTRACNS

This trace record is a continuation of the VCC1 trace record. Up to two VCC2 trace records might follow a VCC1 trace record, depending on the length of the primitive that is traced.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| VCC2 | UP TO 28 BYTES OF ADDITIONAL CNCB DATA |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"VCC2"
**04–1F** Up to 28 bytes of additional CNCB data

# VCDQ entry for work element dequeued

**Entry: VCDQ**
**VIT option:**
 VCNS
**Event:** Work element dequeued
**VIT processing module:**
 ISTRACNS
**Control is returned to:**
 Module invoking the INTRACE macro that caused the record to be produced

This trace record is written whenever a work element is dequeued for processing. The VCDQ trace record identifies the work element being dequeued.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| VCDQ | ID | STATE | CBID | WKEL | WORK ELEMENT TYPE | RU TYPE OR 0 | WORK ELEMENT ADDRESS | CAB ADDRESS OR 0 | 0 | RPH ADDRESS |

**Byte (hex)**
 **Contents**
**00–03** Record ID: C"VCDQ"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** Process state of the associated connection or of the control block processing the dequeued work element:

States when CBID=X'0B' (ISTVLNCB)

**X'00'** Reset state
**X'10'** ACTLINK pending
**X'20'** Active state
**X'30'** DACTLINK pending

States when CBID=X'0C' (ISTPCLCB)

**X'00'** Reset
**X'01'** Logon complete
**X'02'** Logon pending
**X'03'** Logoff pending
**X'04'** Adapter termination pending

States when CBID=X'25' (ISTVCCB)

**X'01'** P1 Ready
**X'02'** P2 DTE waiting
**X'03'** P3 DCE waiting
**X'04'** P4 Data transfer
**X'05'** P5 Call collision
**X'06'** P6 DTE clear request
**X'07'** P7 DCE clear indication
**X'11'** D1 Flow control ready
**X'12'** D2 DTE reset request
**X'13'** D3 DCE reset indication
**X'21'** I1 No interrupt pending
**X'22'** I2 DTE interrupt pending
**X'31'** E1 No error reset
**X'32'** ER CNS error reset
**X'33'** CF CNS cancel forward

States when CBID=X'66' (ISTCAB)

**X'00'** Idle
**X'01'** Connect requested
**X'02'** Connect indicated
**X'03'** Flow control ready
**X'04'** Error reset
**X'05'** Cancel hold
**X'06'** Hold purge
**X'07'** Reset requested
**X'08'** Reset indicated
**X'09'** Error disconnect
**X'0A'** Cleanup
**X'0B'** Disconnect requested
**X'0C'** Disconnect indicated
**X'0D'** Pending error disconnect

States when CBID=X'6B' (ISTCAR)

No states are traced when ISTCAR is the major control block.

States when CBID=X'7B' (ISTRIB)

**X'00'** Idle
**X'01'** Logon in progress
**X'02'** Active
**X'03'** Logoff in progress
**X'04'** DACTLINK in progress
**X'05'** Forced termination in progress

**06** Control block identifier

**X'0B'** ISTVLNCB
**X'0C'** ISTPCLCB
**X'25'** ISTVCCB
**X'66'** ISTCAB

|  |  |  |
|---|---|---|
|  | **X'6B'** | ISTCAR |
|  | **X'7B'** | ISTRIB |
| 07 | If work element is an 'RPL' (see bytes 08–0B) | |
|  | **X'08'** | CONTROL=INQUIRE |
|  | **X'10'** | CONTROL=LOGON |
|  | **X'14'** | CONTROL=TEST |
|  | **X'18'** | CONTROL=XID |
|  | **X'20'** | CONTROL=LOGOFF |
|  | **X'30'** | CONTROL=READ |
|  | **X'40'** | CONTROL=STATUS |
|  | **X'50'** | CONTROL=SET |
|  | **X'60'** | CONTROL=OPEN |
|  | **X'70'** | CONTROL=CLOSE |
|  | **X'80'** | CONTROL=SEND |
|  | **X'90'** | CONTROL=SUSPEND |
|  | **X'A0'** | CONTROL=RESUME |
|  | **X'B0'** | CONTROL=EXPEDITE |
|  | **X'C0'** | CONTROL=CHECK |
|  | **X'D0'** | CONTROL=RESET |
|  | **X'E0'** | CONTROL=SETCPARM |
|  | **X'F0'** | CONTROL=RECEIVE |
|  | **X'FC'** | CONTROL=REPLY |
| 07 | If work element is a 'PICB' (see bytes 08–0B) | |
|  | **X'01'** | LOGON_REQUEST |
|  | **X'02'** | LOGON_CONFIRM |
|  | **X'03'** | LOGOFF_REQUEST |
|  | **X'05'** | CNS_LOGON_REQUEST |
|  | **X'06'** | CNS_LOGON_CONFIRM |
|  | **X'07'** | CNS_LOGOFF_REQUEST |
|  | **X'08'** | CNS_LOGOFF_CONFIRM |
|  | **X'09'** | CNS_LOGOFF_RESPONSE |
|  | **X'0A'** | CNS_LOGOFF_INDICATION |
|  | **X'0B'** | CNS_CLOSEACB_REQUEST |
|  | **X'0C'** | CNS_CLOSEACB_CONFIRM |
|  | **X'0D'** | CNS_CLOSEACB_COMPLETION_INDICATION |
|  | **X'11'** | CONNECT_REQUEST |
|  | **X'12'** | CONNECT_CONFIRM |
|  | **X'13'** | CONNECT_ACCEPT |
|  | **X'14'** | CONNECT_INDICAT |
|  | **X'15'** | DISCONNECT_REQUEST |
|  | **X'16'** | DISCONNECT_INDICAT |
|  | **X'17'** | DISCONNECT_CONFIRM |
|  | **X'18'** | ERROR_DISCONNECT_INDICAT |
|  | **X'19'** | RESTART_INDICAT |
|  | **X'1A'** | CNS_CLEAR |
|  | **X'1B'** | ERROR_DISCONNECT_INDICATION (TIMER) |
|  | **X'21'** | RESET_REQUEST |
|  | **X'22'** | RESET_CONFIRM |
|  | **X'23'** | RESET_ACCEPT |
|  | **X'24'** | RESET_INDICAT |
|  | **X'25'** | ERROR_RESET_INDICAT |
|  | **X'31'** | DATA_OUT (LEVEL1) |
|  | **X'32'** | DATA_IN (LEVEL1) |
|  | **X'33'** | DATA_OUT (LEVEL2) |
|  | **X'34'** | DATA_IN (LEVEL2) |
|  | **X'35'** | INTERRUPT_OUT |

| | |
|---|---|
| **X'36'** | INTERRUPT_CONFIRM |
| **X'37'** | INTERRUPT_INDICATION |
| **X'42'** | STATUS_INDICAT_FINAL_TERMINAT |
| **X'53'** | SUSPEND_REQUEST |
| **X'54'** | RESUME_REQUEST |
| **X'55'** | RNR_IN |
| **X'56'** | RR_IN |
| **X'57'** | ROTATE |
| **X'62'** | TIMER_EVENT |
| **X'63'** | RESTART_VC_TRIGGER |
| **X'71'** | RESET_PENDING_TRIGGER |
| **X'72'** | RESET_COMPLETE_TRIGGER |
| **X'73'** | CLEAR_PENDING_TRIGGER |
| **X'74'** | CLEAR_COMPLETE_TRIGGER |
| **X'75'** | CALL_PENDING_TRIGGER |
| **X'76'** | CALL_COMPLETE_TRIGGER |
| **X'77'** | INTERRUPT_PENDING_TRIGGER |
| **X'78'** | ERROR_RESET_TRIGGER |
| **X'79'** | ERROR_DISCONNECT_TRIGGER |
| **X'82'** | PACKET_IN |
| **X'91'** | TERMINATE_PORT |
| **X'E0'** | CNS_NMVT |
| **X'F0'** | READ_REQ |
| **X'F1'** | READ_CONFIRM |
| **X'F2'** | STATUS_PHYSICAL_REQUEST |
| **X'F3'** | STATUS_PHYSICAL_CONFIRM |
| **X'F4'** | STATUS_VIRTUAL_REQ |
| **X'F5'** | STATUS_VIRTUAL_CONFIRM |

If work element is a 'CNCB' (see bytes 08–0B)

| | |
|---|---|
| **X'01'** | CNS_INQUIRE_REQUEST |
| **X'02'** | CNS_INQUIRE_CONFIRM |
| **X'03'** | CNS_LOGON_REQUEST |
| **X'04'** | CNS_LOGON_CONFIRM |
| **X'05'** | CNS_LOGOFF_REQUEST |
| **X'06'** | CNS_LOGOFF_CONFIRM |
| **X'07'** | CNS_LOGOFF_RESPONSE |
| **X'08'** | CNS_LOGOFF_INDICATION |
| **X'09'** | CNS_CLOSEACB_REQUEST |
| **X'0A'** | CNS_CLOSEACB_CONFIRM |
| **X'0B'** | CNS_CLOSEACB_COMPLETION_INDICATION |

**08–0B**  Work element type:

**C"CNCB"**
   CNCB
**C"RPL"**
   RPL
**C"PICB"**
   PICB
**C"RUPE"**
   RUPE
**C"VCCB"**
   VCCB
**C"CAB"**
   CAB
**C"RIB"**
   RIB

> **C"RPH"**
>> RPH
> **C"TSCB"**
>> TSCB
> **C"UECB"**
>> UECB
> **C"????"**
>> None of the above

**0C–0F** If work element is a 'RUPE', one of the following; otherwise, 0
> **X'0801020A'**
>> ACTLINK
> **X'0801020B'**
>> DACTLINK

**10–13** Work element address

**14–17** If work element is 'PICB', CAB address; otherwise, 0

**18–1B** 0

**1C–1F** Request parameter header (RPH) address

---

# VPST entry for VPOST macro

**Entry: VPST**
**VIT option:**
> PSS
**Event:** VPOST macro
**VIT processing module:**
> ISTRACPS
**Control is returned to:**
> Module invoking the INTRACE macro that caused the record to be produced

This trace record is written when a VTAM module issues a VPOST macro. The VPOST macro generates a system POST event.

| 0 0 0 0<br>0 1 2 3 | 0<br>4 | 0 0 0<br>5 6 7 | 0 0 0 0<br>8 9 A B | 0 0 0 0<br>C D E F | 1 1 1 1 1 1 1 1<br>0 1 2 3 4 5 6 7 | 1 1 1 1<br>8 9 A B | 1 1 1 1<br>C D E F |
|---|---|---|---|---|---|---|---|
| VPST | I D | 0 | ASCB ADDRESS OR 0 | ECB ADDRESS | SUBTASK NAME | ISSUER ADDRESS | COMPLTE CODE OR 0 |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"VPST"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** 0

**08–0B** Address of the ASCB or 0

**0C–0F** Address of the ECB being posted

**10–17** Subtask name

**18–1B** Address of the issuer of the VPOST macro

**1C–1F** Completion code or 0

## VRSM entry for resume after VWAIT

**Entry: VRSM**
**VIT option:**
>   PSS
**event:** VWAIT macro
**VIT processing module:**
>   ISTRACPS

This trace record is written when VTAM resumes control following a system WAIT event generated by the VTAM VWAIT macro.

| 0 0 0 0 | 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| VRSM | I D | 0 | 0 | ECB ADDRESS | SUBTASK NAME | ISSUER ADDRESS | 0 |

**Byte (hex)**
>   **Contents**

**00–03** Record ID: C"VRSM"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–0B** 0
**0C–0F** Address of the ECB that was posted
**10–17** Subtask name
**18–1B** Address of the code that is resuming execution
**1C–1F** 0

## VTAL entry for allocate storage

**Entry: VTAL**
**VIT option:**
>   SMS
**Event:** Allocate storage
**VIT processing module:**
>   ISTRACSM
**Control is returned to:**
>   ISTORMVA

This trace record provides the status of each VTALLOC request issued by VTAM components.

If this entry is associated with an event failure (that is, the return code is nonzero), this entry will be generated whether or not the SMS option is in effect. It is treated as an exception condition and is always traced, regardless of the VIT options specified, if the VIT is active.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VTAL | | | | I D | F L A G | 0 | | STORAGE ADDRESS | | | | SUBPOOL NUMBER OR POOL ID | | | | RETURN ADDRESS | | | | LENGTH OF STORAGE REQUEST | | | | CALLER OF UTILITY OR 0 | | | | RETURN CODE | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"VTAL"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** Flags

**Bit** **Meaning**

**0... ....**
FREED(EXPLICIT) storage request

**1... ....**
FREED(TASK) or FREED(JSTASK) storage request

**06–07** 0

**08–0B** Address of area allocated

**0C–0F** Subpool number of the area allocated

**10–13** Address of the issuer of the VTALLOC macro

**14–17** Length of area allocated

**18–1B** If the VTALLOC macro is issued by a utility routine, this location contains the address of the issuer of the utility routine.

If the VTALLOC macro is not issued by a utility routine, this location contains 0.

**1C–1F** VTALLOC return code

# VTFR entry for free storage

**Entry: VTFR**
**VIT option:**
SMS
**Event:** Free storage
**VIT processing module:**
ISTRACSM
**Control is returned to:**
ISTORMVF

This trace record provides the status of each VTFREE request issued by VTAM components.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VTFR | | | | I D | | 0 | | STORAGE ADDRESS | | | | SUBPOOL NUMBER OR POOL ID | | | | RETURN ADDRESS | | | | LENGTH OF STORAGE FREED | | | | CALLER OF UTILITY OR 0 | | | | 0 | | | |

**Byte (hex)**

      **Contents**

00–03  Record ID: C"VTFR"

04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05–07  0

08–0B  Address of area to be freed

0C–0F  Subpool number of the area allocated

10–13  Address of the issuer of the VTFREE macro

14–17  Length of area to be freed

18–1B  If the VTFREE macro is issued by a utility routine, this location contains the address of the issuer of the utility routine.

      If the VTFREE macro is not issued by a utility routine, this location contains 0.

1C–1F  0

# VWAI entry for VWAIT macro

**Entry:** VWAI

**VIT option:**
      PSS

**Event:** VWAIT macro

**VIT processing module:**
      ISTRACPS

**Control is returned to:**
      Module invoking the INTRACE macro that caused the record to be produced.

This trace record is written when a VTAM module issues a VWAIT macro. The VWAIT macro generates a system WAIT event.

If the "last address in ECB list" field is not 0, then this is a wait on an ECB list. Note that an ECB list is a list of the addresses of ECBs, not a list of ECBs.

| 00 01 02 03 | 04 | 05 06 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|
| VWAI | I D | 0 | LAST ECB ADDRESS OR 0 | ECB ADDRESS | SUBTASK NAME | ISSUER ADDRESS | 0 |

**Byte (hex)**

      **Contents**

00–03  Record ID: C"VWAI"

04      ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

05–07  0

08–0B  Address of the last pointer in an ECB list, or 0. If this field is 0, it implies that the previous field points to an ECB instead of an ECB list.

0C–0F  Address of the ECB that is being waited on, or (if the next field is nonzero) the address of an ECB list

10–17  Subtask name

18–1B  Address of the issuer of the VWAIT macro

> 1C–1F   0

---

# WAIT entry for TPWAIT macro

**Entry:  WAIT**
**VIT option:**
> PSS

**Event:**  TPWAIT macro
**VIT processing module:**
> ISTRACPS

**Control is returned to:**
> ISTAPCTW

This trace record identifies an RPH that has been suspended to wait for some VTAM resource. This entry might be used with the entry immediately preceding it in the table to determine whether the wait is for storage or for a lock. If the preceding entry is for a REQSTORE, a waiting-for-storage condition may exist. If the preceding entry is for a lock request, the wait can be caused by that request. However, VTAM frequently uses a TPWAIT/TPPOST to synchronize intercomponent functions. Therefore, the TPWAIT entry does not necessarily identify an unusual condition.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WAIT | | | | ID | 0 | FLAGS | | PST ADDRESS | | | | PAB ADDRESS | | | | RETURN ADDRESS | | | | WORK EL Q OR NEXT DSP Q LVL | | | | MODULE NAME OR DVT ADDRESS | | | | RPH ADDRESS | | | |

**Byte (hex)**
> **Contents**

**00–03**   Record ID: C"WAIT"
**04**   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05**   0
**06**   PAB flag field (PABFLAGS)

> **Bit    Meaning**
> **1... ....**
> > PAB is unconditionally scheduled.
>
> **.1.. ....**
> > PAB closedown is in progress.
>
> **..1. ....**
> > PAB is synchronous.
>
> **...1 ....**
> > PAB extension is present.
>
> **.... 1...**
> > Do not dequeue work element.
>
> **.... .1..**
> > Do not detach the RPH.
>
> **.... ..1.**
> > Indicates a very extended PAB.

**.... ...1**
> Indicates a slightly extended PAB.

**07** PAB flag field (PABFLGS1)

> **Bit** **Meaning**
>
> **1... ....**
> > Switch the PST address of this PAB's major control block to the new PST address contained in DYPNWPST.
>
> **.1.. ....**
> > This PAB has a data space extension.
>
> **..1. ....**
> > This PAB's major control block is an FMCB.
>
> **...1 ....**
> > PAB can be referenced in PSW disable mode.
>
> **.... 1...**
> > PAB is persistent.
>
> **.... .1..**
> > APSTERM/APSINIT FMCB during PAB dispatch.
>
> **.... ..xx**
> > Reserved.

**08–0B** PST address

**0C–0F** PAB address

**10–13** Address of the issuer of the TPWAIT macro

**14–17** PAB work element queue or next dispatchable queue level for a very extended PAB

**18–1B** Module name abbreviation (bytes 4, 5, 7, and 8 of the name of the next module to get control), PAB DVT address (high order bit of X'18' = 0), function, or process.

**1C–1F** Request parameter header (RPH) address

# XBA1 entry for extended buffer list (Part 1)

**Entry: XBA1**
**VIT option:**
> XBUF

**Event:** OPTCD=XBUFLST request on APPCCMD macroinstruction
**VIT processing module:**
> ISTITCXB

**Control is returned to:**
> ISTPSCFM, ISTPSCFR, ISTPSCFS, or ISTPSCMU

The XBA1 record is created by any of the following conditions:
- An HPDT send is issued (send request with OPTCD=XBUFLST specified).
- An HPDT send is completed and RPLXSRV is not set.
- An HPDT receive (APPCCMD CONTROL=RECEIVE with OPTCD=XBUFLST) is completed.

The entry contains information required to correlate this set of extended buffer list entries to an appropriate APPC entry (either ACA1 or ACP1/ACR1).

The number of trace records required to represent an application request buffer list is variable based on the number of entries present in the buffer list.

The first record in a request is XBA1. This record is followed with as many XBA2 and XBA3 record pairs as is necessary to record the entire application buffer list contents.

There is no limit to the number of trace entries that can be produced. Although VTAM groups records together as a single entity in terms of recording in the trace recording media, there is a finite limit to how many records can be written externally or recorded internally at one time. Due to multiprocessing, other VIT records from unrelated operations can be interleaved in the trace output between the XBUF entries for a given operation. To provide correlation between groups of XBUF entries, the RPL address is traced in each entry.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B | C D E F |
| XBA1 | I D  0 | RPL ADDRESS | 0 | RPH ADDRESS |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"XBA1"

**04**  ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07**  0

**08–0B**  RPL address. This field corresponds to the RPL address of the ACA1, ACP1, or ACR1 entry.

**0C–1B**  0

**1C–1F**  RPH address

# XBA2 entry for extended buffer list (Part 2)

**Entry:  XBA2**

**VIT option:**

XBUF

**Event:**  OPTCD=XBUFLST request on APPCCMD macroinstruction

**VIT processing module:**

ISTITCXB

This trace record is a continuation of the XBA1 entry. It contains information about an XBUFLST entry.

| 0 0 0 0 | 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 | 8 | 9 | A | B | C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| XBA2 | RPL ADDRESS | 0 | S O U R C E | T Y P E | F L A G | CSM BUFFER TOKEN | CSM DATA SPACE ALET | ADDRESS OF DATA |

**Byte (hex)**

**Contents**

**00–03**  Record ID: C"XBA2"

**04–07**  RPL address

**08**  0

| | | |
|---|---|---|
| **09** | Buffer source | |
| | **80** | ECSA |
| | **40** | Data space |
| | **20** | User data space |
| | **10** | User storage area other than a data space |
| **0A** | Buffer type | |
| | **80** | Fixed |
| | **40** | Pageable |
| | **20** | Eligible to be made pageable |
| **0B** | BLXEN_FLAGS flag byte | |
| **0C–17** | CSM buffer token | |
| **18–1B** | CSM data space ALET | |
| **1C–1F** | Address of data | |

# XBA3 entry for extended buffer list (Part 3)

**Entry: XBA3**
**VIT option:**
    XBUF
**Event:** OPTCD=XBUFLST request on APPCCMD macroinstruction
**VIT processing module:**
    ISTITCXB

This trace record is a continuation of the XBA2 entry.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F | 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| XBA3 | RPL ADDRESS | APPL SUPPLED LENGTH | VTAM ACCPTED LENGTH | 0 |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"XBA3"
**04–07** RPL address
**08–0B** Length of data provided by application
**0C–0F** Length of data accepted by VTAM (send completion only) or 0
**10–31** 0

# XBI1 entry for extended buffer list AMU, SPAC or TSCB

**Entry: XBI1**
**VIT option:**
    XBUF
**Event:** Extended buffer list AMU, SPAC, or TSCB
**VIT processing module:**
    ISTITCXB
**Control is returned to:**
    INTRACE invoker

This record is produced when an extended buffer list APPC message unit (AMU), ShortPAC (SPAC), or transmission subsystem control block (TSCB) is built by VTAM.

The number of trace records required to represent a VTAM internal extended buffer list is variable based on the number of entries in the structure.

The first record in a request is XBI1. This record is followed with as many XBI2 and XBI3 record pairs as is necessary to record the entire VTAM internal extended buffer list contents. The first XBI2 and XBI3 record pair contains information relating to the extended buffer list control entry. Subsequent XBI2 and XBI3 record pairs contain information relating to extended buffer list data entries. In addition, if the internal extended buffer list spans more than one AMU, SPAC, or TSCB, then a new XBI1 record is recorded at the point at which a new AMU, SPAC, or TSCB is used. In this case, the XBI1 record contains the previous AMU, SPAC, or TSCB address so that the relationship between this record and the previous set of records can be maintained.

There is no limit to the number of trace entries that can be produced. Although VTAM groups records together as a single entity in terms of recording in the trace recording media, there is a finite limit to how many records can be written externally or recorded internally at one time. Due to multiprocessing, other VIT records from unrelated operations can be interleaved in the trace output between the XBUF entries for a given operation. To provide correlation between groups of XBUF entries, the AMU, SPAC, or TSCB address is traced in each entry.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XBI1 | | | | ID | 0 | | | TSCB OR AMU ADDRESS | | | | 0 OR PRVIOUS TSCB OR AMU ADDRESS | | | | RETURN ADDRESS | | | | 0 | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**
  **Contents**
**00–03** Record ID: C"XBI1"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07** 0
**08–0B** TSCB, SPAC, or AMU address
**0C–0F** Address of the previous TSCB, SPAC, or AMU address, or zeros if this is the first TSCB, SPAC, or AMU.
**10–13** Return address
**14–1B** 0
**1C–1F** RPH address

## XBI2 entry for extended buffer list AMU, SPAC, or TSCB

**Entry: XBI2**
**VIT option:**
    XBUF
**Event:** Extended buffer list AMU, SPAC, or TSCB
**VIT processing module:**
    ISTITCXB

This trace record is a continuation of the XBI1 entry. This record contains information about an extended buffer list entry contained in an AMU, SPAC, or

TSCB.

| 0 0 0 0 | 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 | 9 | A | B | C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| XBI2 | TSCB OR AMU ADDRESS | V E R S I O N | S O U R C E | T Y P E | 0 | BUFFER TOKEN OR 0 | CSM DATA SPACE ALET | ADDRESS OF DATA |

**Byte (hex)**
  **Contents**
**00–03** Record ID: C"XBI2"
**04–07** XBUFLST TSCB, SPAC, or AMU address
**08** Version of CSM buffer descriptor
**09** Buffer source
  **80** ECSA
  **40** Data space
  **20** User data space
  **10** User storage area other than a data space
**0A** Buffer type
  **80** Fixed
  **40** Pageble
  **20** Eligible to be made pageable
**0B** 0
**0C–17** Buffer token or zeros
**18–1B** CSM data space ALET
**1C–1F** Address of data

# XBI3 entry for extended buffer list AMU, SPAC, or TSCB
**Entry: XBI3**
**VIT option:**
  XBUF
**Event:** Extended buffer list AMU, SPAC, or TSCB
**VIT processing module:**
  ISTITCXB

This trace record is a continuation of the XBI1 entry. This record contains additional information about an extended buffer list entry contained in an AMU, SPAC, or TSCB.

| 00 01 02 03 | 04 05 06 07 | 08 09 0A 0B | 0C 0D 0E 0F | 10 11 12 13 | 14 | 15 16 17 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|
| XBI3 | TSCB OR AMU ADDRESS | LENGTH OF DATA | VTAM IO BUFFER ADDRESS OR 0 | STORAGE RELEASE ROUTINE | X B F L F L G S | 0 |

**Byte (hex)**
> **Contents**
> 00–03   Record ID: C"XBI3"
> 04–07   XBUFLST TSCB, SPAC, or AMU address
> 08–0B   Length of data pointed to by this entry
> 0C–0F   VTAM I/O buffer address if entry represents data contained in I/O buffer or zeros
> 10–13   Storage release routine
> 14   XBFLFLGS byte
> 15–1F   0

# XCFC entry for save message (Part 1)

**Entry:**  **XCFC**
**VIT option:**
> XCF
**Event:**  A message from another XCF member is saved or a saved message is processed.
**VIT processing module:**
> ISTITCXF
**Control is returned to:**
> ISTTSCMX or ISTTSCDX

The XCFC trace record is written when VTAM cannot immediately receive a message from another VTAM because of a resource shortage or when VTAM is processing a previously saved message.

| 00 01 02 03 | 04 | 05 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 | 18 19 | 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|
| XCFC | ID | 0 | C O D E | ORIGINAL MESSAGE TOKEN | 0 | # ENT | RPH ADDRESS |

**Byte (hex)**
> **Contents**
> 00–03   Record ID: C"XCFC"
> 04   ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
> 05–06   0
> 07   Request code:

C"D"    VTAM has discarded the received message.
C"Q"    VTAM has sent a query to determine how many saved messages exist.
C"R"    VTAM is redelivering a saved message to the message exit.
C"S"    VTAM's first attempt to receive a message failed, so the message has been saved.

**08–17** Original message token value
**18–19** 0
**1A–1B** Number of entries if Request Code=C'Q'; otherwise 0.
**1C–1F** Request parameter header (RPH) address

## XCC2 entry for save message (Part 2)

**Entry:  XCC2**
**VIT option:**
    XCF
**Event:**  A message from another XCF member is saved
**VIT processing module:**
    ISTITCXF

This trace record is a continuation of the XCFC entry. It shows additional information about a saved message.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XCC2 | | | | 0 | | | C O D E | SAVED MESSAGE TOKEN | | | | | | | | | | | | | | | | REASON CODE | | | | RETURN CODE | | | |

**Byte (hex)**
    **Contents**
**00–03** Record ID: C"XCC2"
**04–06** 0
**07** Request code
**08–17** Saved message token value
**18–1B** Reason code from an MVS IXCMSGC macroinstruction
**1C–1F** Return code from an MVS IXCMSGC macroinstruction

Refer to *z/OS MVS Programming: Sysplex Services Reference* for a description of the return codes and reason codes.

## XCFJ entry for join XCF group (Part 1)

**Entry:  XCFJ**
**VIT option:**
    XCF
**Event:**  VTAM joins an XCF group
**VIT processing module:**
    ISTITCXF
**Control is returned to:**
    ISTTSCIX for XCF connectivity

    ISTFSUXJ for multiple node persistent session or coupling facility structure support.

The XCFJ trace record is written when VTAM joins an XCF group. It shows information about the XCF connection established by an MVS IXCJOIN macroinstruction, placing an XCF member in an active state and associating it with an XCF group.

| 0 0 | 0 0 | 0 | 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 | 2 3 | 4 | 5 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| XCFJ | | I D | 0 | XCF TOKEN | MAJOR CONTROL BLOCK ADDRESS | XCF GROUP NAME | RPH ADDRESS |

**Byte (hex)**

> **Contents**

**00–03** Record ID: C"XCFJ"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05–07** 0

**08–0F** XCF token value

**10–13** Major control block address of:
- ISTXCFCB for XCF connectivity.
- 0 for coupling facility structure support.

**14–1B** XCF group name:
- ISTXCF*vv* for XCF connectivity, where *vv* is the group ID specified on the XCFGRPID start option. If a XCFGRPID value was not specified, the group name is ISTXCF.
- ISTCFS*vv* for coupling facility structure support, where *vv* is the group ID specified on the XCFGRPID start option. If a XCFGRPID value was not specified, the group name is ISTCFS01.

**1C–1F** Request parameter header (RPH) address

# XCJ2 entry for join XCF group (Part 2)

**Entry: XCJ2**

**VIT option:**

> XCF

**Event:** VTAM joins an XCF group

**VIT processing module:**

> ISTITCXF

This trace record is a continuation of the XCFJ entry. It shows additional information about the XCF connection established by an MVS IXCJOIN macroinstruction.

| 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 | 3 4 5 6 7 8 9 A B C D E F 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F |
| XCJ2 | GROUP MEMBER NAME | S Y S C L O N E | V A L U E | 0 | REASON CODE | RETURN CODE |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"XCJ2"

**04–13** Group member name comprises:
- Control point name
- Network identifier

**14–15** MVS &SYSCLONE value for XCF connectivity

0 for multiple node persistent session support

The MVS &SYSCLONE value is defined in MVS. For complete information about how the symbol is defined, refer to *z/OS MVS Setting Up a Sysplex* and *z/OS MVS Initialization and Tuning Reference*.

**16–17** 0

**18–1B** Reason code from an MVS IXCJOIN macroinstruction

**1C–1F** Return code from an MVS IXCJOIN macroinstruction

Refer to *z/OS MVS Programming: Sysplex Services Reference* for a description of the return codes and reason codes.

## XCFL entry for leave XCF group (Part 1)

**Entry: XCFL**

**VIT option:**
XCF

**Event:** VTAM leaves an XCF group

**VIT processing module:**
ISTITCXF

**Control is returned to:**
ISTDECDH

The XCFL trace record is written when VTAM leaves an XCF group. It shows information about the XCF connection terminated by an MVS IXCLEAVE macroinstruction, placing an XCF member in an undefined state and disassociating it from its XCF group.

| 0 0 0 0 | 0 | 0 0 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 6 7 | 8 9 A B C D E F | 0 1 2 3 | 4 5 6 7 8 9 A B | C D E F |
| XCFL | I D | 0 | XCF TOKEN | XCFCB ADDRESS | XCF GROUP NAME | RPH ADDRESS |

**Byte (hex)**
  **Contents**
**00–03** Record ID: C"XCFL"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05–07** 0
**08–0F** XCF token value
**10–13** XCF control block address (ISTXCFCB)
**14–1B** XCF group name
**1C–1F** Request parameter header (RPH) address

# XCL2 entry for leave XCF group (Part 2)

**Entry: XCL2**
**VIT option:**
  XCF
**Event:** VTAM leaves an XCF group
**VIT processing module:**
  ISTITCXF

This trace record is a continuation of the XCFL entry. It shows additional information about the XCF connection established or terminated by an MVS IXCLEAVE macroinstruction.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 | 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 | 4 5 | 6 7 | 8 9 A B | C D E F |
| XCL2 | GROUP MEMBER NAME | S Y S C L O N E  V A L U E | 0 | REASON CODE | RETURN CODE |

**Byte (hex)**
  **Contents**
**00–03** Record ID: C"XCL2"
**04–13** Group member name comprises:
  • Control point name
  • Network identifier
**14–15** MVS &SYSCLONE value

  The MVS &SYSCLONE value is defined in MVS. For complete information about how the symbol is defined, refer to *z/OS MVS Setting Up a Sysplex* and *z/OS MVS Initialization and Tuning Reference*.
**16–17** 0

**18–1B** Reason code from an MVS IXCLEAVE macroinstruction

**1C–1F** Return code from an MVS IXCLEAVE macroinstruction

Refer to *z/OS MVS Programming: Sysplex Services Reference* for a description of the return codes and reason codes.

## XCFM entry for XCF group member (Part 1)

**Entry: XCFM**

**VIT option:**
    XCF

**Event:** A member joins or leaves an XCF group

**VIT processing module:**
    ISTITCXF

**Control is returned to:**
    ISTTSCGX or ISTTSCMX for XCF connectivity. ISTFSXGE for multiple node persistent session support

The XCFM trace record is written when an adjacent VTAM joins or leaves an XCF group, or when an initial message is received from an adjacent VTAM after this VTAM joins the group.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XCFM | | | | ID | 0 | O S T A T E | N S T A T E | XCF TOKEN | | | | | | | | AMRU ADDRESS OR 0 | | | | XCF GROUP NAME | | | | | | | | RPH ADDRESS | | | |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"XCFM"

**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.

**05** 0

**06** Old event state when a member joins or leaves the group

    0 when an initial message is received

    The event states are defined in the group exit parameter list (IXCYGEPL). Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of the event states.

**07** New event state when a member joins or leaves the group

    0 when an initial message is received

    The event states are defined in the group exit parameter list (IXCYGEPL). Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of the event states.

**08–0F** XCF token value

**10–13** AMRU address for XCF connectivity. 0 for coupling facility support

**14–1B** XCF group name:

> • ISTXCF*vv* for XCF connectivity, where *vv* is the group ID supplied on
> the XCFGRPID start option. If XCFGRPID was not specified, the group
> name is ISTXCF.
> • ISTCFS*vv* for coupling facility structure support, where *vv* is the group
> ID supplied on the XCFGRPID start option. If XCFGRPID was not
> specified, the group name is ISTCFS01.

**1C–1F** Request parameter header (RPH) address

# XCM2 entry for XCF group member (Part 2)

**Entry: XCM2**
**VIT option:**
>    XCF
**Event:** A member joins or leaves an XCF group
**VIT processing module:**
>    ISTITCXF

This trace record is a continuation of the XCFM entry. It shows additional
information about the XCF member when an adjacent VTAM joins or leaves an
XCF group, or when an initial message is received from an adjacent VTAM after
this VTAM joins the group.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 | 1 1 | 1 | 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 | 4 5 | 6 | 7 8 9 A B C D E F |
| XCM2 | GROUP MEMBER NAME | S Y S C L O N E | V A L U E | E T Y P E | 0 |

**Byte (hex)**
>    **Contents**
**00–03** Record ID: C"XCM2"
**04–13** Group member names comprises:
> • Control point name
> • Network identifier

**14–15** MVS &SYSCLONE value for XCF connectivity

>    0 for coupling facility structure support

>    The MVS &SYSCLONE value is defined in MVS. For complete information
>    about how the symbol is defined, refer to *z/OS MVS Setting Up a Sysplex*
>    and *z/OS MVS Initialization and Tuning Reference*.

**16** Event type when a member joins or leaves the group

>    0 when an initial message is received

>    The event type is defined by the group exit parameter list (IXCYGEPL).
>    Refer to *z/OS MVS Data Areas, Vol 3 (IVT-RCWK)* for an explanation of the
>    event types.

**17–1F** 0

## XCFR entry for receive message (Part 1)

**Entry: XCFR**
**VIT option:**
> XCF

**Event:** A message is received from another XCF group member.
**VIT processing module:**
> ISTITCXF

**Control is returned to:**
> ISTTSCMX, ISTTSCBX, or ISTFSXME

The XCFR trace record is written when VTAM receives a message from another active member of an XCF group. It shows information about the message when an MVS IXCMSGI macroinstruction is invoked.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| XCFR | | | | ID | 0 | FLAGS | STATE | XCF TOKEN | | | | | | | | 0 | TYPE | LENGTH | | XFNCB ADDRESS | | | | DATA ELEMENT ADDRESS | | | | RPH ADDRESS | | | |

**Byte (hex)**
> **Contents**

**00–03** Record ID: C"XCFR"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Flag field from MSGCNTRL area
**07** State of the link if user type=C'X'; otherwise, 0.
**08–0F** XCF token value
**10** 0
**11** User type:

> **C'X'** VTAM/XCF

> **C'M'** MNPS

**12–13** Length of message
**14–17** XFNCB address if user type is C'X'; otherwise, 0.
**18–1B** Data element address
**1C–1F** Request parameter header (RPH) address

## XCR2 entry for receive message (Part 2)

**Entry: XCR2**
**VIT option:**
> XCF

**Event:** A message is received from another XCF group member.
**VIT processing module:**
> ISTITCXF

This trace record is a continuation of the XCFR entry. It is written when an MVS IXCMSGI macroinstruction is invoked. It shows the return code and reason code from the macroinstruction.

### Mapping for Usertype=C'M'

When Usertype (byte 6) is C'M', the mapping for XCR2 is as follows:

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B C D E F | 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| XCR2 | 0 | T Y P E | S T A T U S | APPL NAME | 0 | REASON CODE | RETURN CODE |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"XCR2"

**04–05** 0

**06** User type:

**C'M'** MNPS

**07** Message reply status

**08–0F** Application name

**10–17** 0

**18–1B** Reason code from an MVS IXCMSGI macroinstruction

**1C–1F** Return code from an MVS IXCMSGI macroinstruction

### Mapping for Usertype=C'X'

When Usertype (byte 6) is C'X', the mapping for XCR2 is as follows:

| 0 0 0 0 | 0 0 | 0 | 0 | 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 5 | 6 | 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| XCR2 | 0 | T Y P E | 0 | MESSAGE TOKEN | REASON CODE | RETURN CODE |

**Byte (hex)**

**Contents**

**00–03** Record ID: C"XCR2"

**04–05** 0

**06** User type:

**C'X'** VTAM/XCF

**07** 0

**08–17** Message token

**18–1B** Reason code from an MVS IXCMSGI macroinstruction

**1C–1F** Return code from an MVS IXCMSGI macroinstruction

Refer to *z/OS MVS Programming: Sysplex Services Reference* for a description of the return codes and reason codes.

# XCFS entry for send message (Part 1)

**Entry:** XCFS
**VIT option:**
XCF
**Event:** A message is sent from this VTAM to another XCF group member
**VIT processing module:**
ISTITCXF
**Control is returned to:**
ISTTSCOX or ISTFSPFN

The XCFS trace record is written when VTAM sends a message to another active member of an XCF group. It shows information about the message when an MVS IXCMSGO macroinstruction is invoked.

| 0 0 0 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 0 | 1 | 1 | 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F | 0 | 1 | 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| XCFS | I D | 0 | F L A G S | S T A T E | XCF TOKEN | 0 | T Y P E | L E N G T H | XFNCB ADDRESS | DATA ELEMENT ADDRESS | RPH ADDRESS |

**Byte (hex)**
**Contents**
**00–03** Record ID: C"XCFS"
**04** ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
**05** 0
**06** Flag field from MSGCNTRL area
**07** State of the link if user type=C'X'; otherwise, 0.

**Note:** If the XFNCB address is 0, the link state is also 0.
**08–0F** XCF token value
**10** 0
**11** User type:

**C'X'** VTAM/XCF

**C'M'** MNPS
**12–13** Length of message
**14–17** XFNCB address if user type =C'X'; otherwise, 0.
**18–1B** Data element address if user type =C'X'; otherwise, 0.
**1C–1F** Request parameter header (RPH) address

# XCS2 entry for send message (Part 2)

**Entry:** XCS2
**VIT option:**
XCF
**Event:** A message is sent from this VTAM to another XCF group member
**VIT processing module:**
ISTITCXF

This trace record is a continuation of the XCFS entry. It is written when an MVS IXCMSGO macroinstruction is invoked and either the return code is greater than 0 or the message token was obtained from XCF by VTAM request on the macroinstruction. It shows the return code and reason code from the macroinstruction, and the message token if obtained by VTAM.

## Mapping for Usertype=C'M'

When Usertype (byte 6) is C'M', the mapping for XCS2 is as follows:

| 00 01 02 03 | 04 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F | 10 11 12 13 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|---|
| XCS2 | 0 | TYPE | STATUS | APPL NAME | 0 | REASON CODE | RETURN CODE |

**Byte (hex)**

    **Contents**

**00–03**   Record ID: C"XCS2"

**04–05**   0

**06**   User type:

    **C'M'**   MNPS

**07**   Message reply status if usertype=C'M'.

**08–0F**   Application name

**10–17**   0

**18–1B**   Reason code from an MVS IXCMSGO macroinstruction

**1C–1F**   Return code from an MVS IXCMSGO macroinstruction

## Mapping for Usertype=C'X'

When Usertype (byte 6) is C'X', the mapping for XCS2 is as follows:

| 00 01 02 03 | 04 05 | 06 | 07 | 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 | 18 19 1A 1B | 1C 1D 1E 1F |
|---|---|---|---|---|---|---|
| XCS2 | 0 | TYPE | 0 | MESSAGE TOKEN OR 0 | REASON CODE | RETURN CODE |

**Byte (hex)**

    **Contents**

**00–03**   Record ID: C"XCS2"

**04–05**   0

**06**   User type:

    **C'X'**   VTAM/XCF

**07**   0

**08-17**   Message token or 0

18-1B    Reason code from an MVS IXCMSGO macroinstruction
1C–1F    Return code from an MVS IXCMSGO macroinstruction

Refer to *z/OS MVS Programming: Sysplex Services Reference* for a description of the return codes and reason codes.

# XCFX entry for VTAM XCF message exit or notify exit driven

**Entry:  XCFX**
**VIT option:**
        XCF
**Event:**  VTAM XCF message exit or notify exit entered
**VIT processing module:**
        ISTITCXF
**Control is returned to:**
        ISTTSCMX (message exit), ISTTSCFX (notify exit), or ISTFSXME

The XCFX trace record is written when either the VTAM XCF message exit or notify exit is entered.

| 0 0 | 0 1 | 0 2 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 | 0 9 | 0 A | 0 B | 0 C | 0 D | 0 E | 0 F | 1 0 | 1 1 | 1 2 | 1 3 | 1 4 | 1 5 | 1 6 | 1 7 | 1 8 | 1 9 | 1 A | 1 B | 1 C | 1 D | 1 E | 1 F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XCFX | | | | ID | 0 | F L A G | C O D E | ORIGINAL MESSAGE TOKEN | | | | | | | | | | | | | | | | SOURCE MESSAGE CODE | | | | | | | |

**Byte (hex)**
        **Contents**
01–03    Record ID: C'XCFX'
04       ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'.
05       0
06       Flag Field (Message exit)

        **Bit        Meaning**
        **B'1... ....'**
                Message exit was solicited by the user.
        **B'.1.. ....'**
                Sender requested that XCF manage the response to this message.
        **B'..1. ....'**
                This message is a response being managed by XCF.
        **B'...1 ....'**
                Some portion of the message was delivered by the message-in service.
        **B'.... 1...'**
                Message was saved with message control SAVEMSG service.
        **B'.... .1..'**
                Sender requested ordered message delivery.
        **B'.... ..X.'**
                0
        **B'.... ...1'**
                Additional data presented to the message exit in message exit extension.

Flag Field (Notify exit)

**Bit**  **Meaning**

**B'1... ....'**

Sender specified SENDTO(GROUP) on the IXCMSGO invocation.

**B'.1.. ....'**

Sender requested that XCF manage the response to this message.

**B'..1. ....'**

This message is a response being managed by XCF.

**B'...1 ....'**

Desired sends not initiated by the message-out service.

**B'.... 1...'**

Expected response not received.

**B'.... .1..'**

Message considered to be complete.

**B'.... ..1.'**

Message did not complete within the timeout period.

**B'.... ...1'**

Message was canceled before normal completion occurred.

**07**  Function Code

C'C' - Message exit for CFS

C'M' - Message exit for VTAM/XCF

C'N' - Notify exit for VTAM/XCF

**08–17**  Original Message token

**18–1F**  Source message code

# XID entry for exchange ID (Part 1)

**Entry:**  XID

**VIT option:**

CIA

**Event:**  Exchange ID (Part 1)

**VIT processing module:**

ISTRACCI

**Control is returned to:**

ISTTSCCU or ISTTSC8X

This trace record contains information about inbound or outbound exchange ID processing.

| 0 0 0 0 | 0 0 0 0 | 0 0 0 0 | 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
|---|---|---|---|
| 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| XIDI XIDO | NCB ADDRESS | BPB ADDRESS OR 0 | XID DATA |

**Byte (hex)**

**Contents**

**00–03**  Record ID:

C"XIDI" for inbound XID

C"XIDO" for outbound XID

**04–07**  NCB address

**08–0B**  Boundary physical unit block (BPB) address or 0

0C–1F   20 bytes of XID data

## XID2 entry for exchange ID (Part 2)

**Entry:  XID2**
**VIT option:**
        CIA
**Event:**  Exchange ID (Part 2)
**VIT processing module:**
        ISTRACCI

This trace record is a continuation of the XID trace record.

| 0 0 0 0 | 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 2 3 | 4 5 6 7 8 9 A B C D E F 0 1 2 3 4 5 6 7 8 9 A B C D E F |
| XID2 | ADDITIONAL XID DATA |

**Byte (hex)**
        **Contents**
**00–03**   Record ID: C"XID2"
**04–1F**   28 bytes of additional XID data

## XPST entry for TPSWPST macro

**Entry:  XPST**
**VIT option:**
        PSS
**Event:**  TPSWPST macro
**VIT processing module:**
        ISTRACPS
**Control is returned to:**
        ISTAPCSP

This trace record identifies a VTAM process (PAB) that is switching PSTs during termination of a task. Switching PSTs causes the PAB to be scheduled for dispatch under the new PST.

| 0 0 0 0 | 0 | 0 | 0 0 | 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
| 0 1 2 3 | 4 | 5 | 6 7 | 8 9 A B | C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| XPST | I D | 0 | F L A G S | NEW PST ADDRESS | PAB ADDRESS | ISSR | WEQ | MOD NAME OR DVT ADDRESS | RPH ADDRESS OR 0 |

**Byte (hex)**
        **Contents**
**00–03**   Record ID: "XPST"

|     |     |
| --- | --- |
| **04** | ID is the primary address space ID (ASID). This field is 0 if the ASID is greater than X'FF'. |
| **05** | 0 |
| **06** | PAB flag field (PABFLAGS) |

| Bit | Meaning |
| --- | --- |
| **1... ....** | |
| | PAB is unconditionally scheduled. |
| **.1.. ....** | |
| | PAB closedown is in progress. |
| **..1. ....** | |
| | PAB is synchronous. |
| **...1 ....** | |
| | PAB extension is present. |
| **.... 1...** | |
| | Do not dequeue work element. |
| **.... .1..** | |
| | Do not detach the RPH. |
| **.... ..1.** | |
| | Indicates a very extended PAB. |
| **.... ...1** | |
| | Indicates a slightly extended PAB. |

|     |     |
| --- | --- |
| **07** | PAB flag field (PABFLGS1) |

| Bit | Meaning |
| --- | --- |
| **1... ....** | |
| | Switch the PST address of this PAB's major control block to the new PST address contained in DYPNWPST. |
| **.1.. ....** | |
| | This PAB has a data space extension. |
| **..1. ....** | |
| | This PAB's major control block is an FMCB. |
| **...1 ....** | |
| | PAB can be referenced in PSW disable mode. |
| **.... 1...** | |
| | PAB is persistent. |
| **.... .1..** | |
| | APSTERM/APSINIT FMCB during PAB dispatch. |
| **.... ..xx** | |
| | 0 |

|     |     |
| --- | --- |
| **08–0B** | New PST address under which this PAB will now be dispatched |
| **0C–0F** | PAB address |
| **10–13** | Address of the issuer of the TPSWPST macro |
| **14–17** | PAB work element queue |
| **18–1B** | Module name abbreviation or PAB DVT address |
| **1C–1F** | Request parameter header (RPH) address or 0 |

# Appendix B. Internal topology traces

The internal topology traces provide a record of the creation, update, and deletion of TRS (topology and routing services) topology records. Trace data for the internal topology traces is always automatically recorded.

There are three locations where topology tracing is done:
- In a NDREC (node record) trace table following the NDREC control block, where the creation and update of a node record is recorded.
- In a TGREC (TG record) trace table following the TGREC control block, where the creation and update of a TG record is recorded.
- In a common TRS trace table, where the deletion of NDRECs and TGRECs are recorded.

## The NDREC trace table

The NDREC trace table contains entries describing the creation and update of a node record. It is located following the CV45 in a NDREC. It is pointed to by a trace pointer in the NDREC. The first 8 bytes of the NDREC trace table is the header, which contains the following information:

**Byte (hex)**
> **Contents**

**00–03** The current RSN (resource sequence number) of the last update to this NDREC. The current RSN is used when recording TDUs (topology database updates) related to the NDREC. Because duplicate TDU information can be received from multiple adjacent nodes, the RSN in the TDU is compared to the current RSN in the NDREC trace. The TDU will only be recorded in the NDREC trace if the RSN in the TDU is greater than the current RSN in the NDREC trace, indicating that the information in the TDU is more recent than that in the NDREC.

**04–07** Address of the current NDREC trace table entry. This is the last trace entry where data was recorded.

Following the header are three NDREC trace table entries. The information about the creation of the NDREC is recorded in the first NDREC trace entry. This entry remains unchanged for the life of the NDREC. Recording of trace events will continue in the remaining two trace table entries, with the trace wrapping back to the second entry after the third entry is completed.

## NDREC trace record descriptions

This topic contains the NDREC trace record descriptions. The entries are listed alphabetically by entry name.

### NDREC CPC entry for CP-CP session changes

> **Entry:** CPC
> **Event:** CP-CP session changes
> **Modules Generating Entry:**
> > ISTTRPCS

**409**

These trace records are issued when the status of a CP-CP session with this node changes.

| 0 0 0 | 0 | 0 0 0 | 0 | 0 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 | 7 | 8 | 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| CPC | A C T I O N | V4580 FLAGS | N D T I M E | F L A G S | 0 | V4580 RSN | TIME- STAMP |

**Byte (hex)**
>    **Contents**
**00–02**  Record ID: C"CPC"
**03**    Action

>    • C"U" Update CP-CP session status

**04–06**  Flag bytes from the CV4580
**07**    Number of days left before NDREC is garbage collected
**08**    Flag byte

>    **Bit**    **Meaning**
>    **11.. ....**
>        Node type from NDREC

>        **B′00′**  End node

>        **B′01′**  Network node

>        **B′10′**  Virtual node

>    **..1. ....**
>        Garbage collection indicator from NDREC
>    **...1 ....**
>        Adjacent node indicator from NDREC
>    **.... 1...**
>        DLUR end node indicator from NDREC

**09–17**  0
**18–1B**  Resource sequence number from CV4580
**1C–1F**  First word of timestamp for trace entry

# NDREC HST entry for host node create
**Entry:** **HST**
**Event:** Host node record creation
**Modules Generating Entry:**
>    ISTTRCIT

This trace record is issued when the node record for the host node is created.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| HST | | | ACTION | V4580 FLAGS | | | NDTIME | FLAGS | 0 | | | | | | | | | | | | | | | V4580 RSN | | | | TIME-STAMP | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–02 | Record ID: C"HST" |
| 03 | Action |

• C"C" Create host node record

| | |
|---|---|
| 04–06 | Flag bytes from the CV4580 |
| 07 | Number of days left before NDREC is garbage collected |
| 08 | Flag byte |

**Bit**     **Meaning**

**11.. ....**

         Node type from NDREC

         **B′00′**     End node

         **B′01′**     Network node

         **B′10′**     Virtual node

| | |
|---|---|
| 09–17 | 0 |
| 18–1B | Resource sequence number from CV4580 |
| 1C–1F | First word of timestamp for trace entry |

# NDREC MOD entry for MODIFY TOPO command

**Entry: MOD**
**Event:** MODIFY TOPO operator command
**Modules Generating Entry:**
     ISTTROMT

This trace record is issued when the MODIFY TOPO command with the FUNCTION=QUIESCE or FUNCTION=NORMAL operand is entered by an operator.

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| MOD | | | ACTION | V4580 FLAGS | | | NDTIME | FLAGS | SCOPE | 0 | | | | | | | | | | | | | | V4580 RSN | | | | TIME-STAMP | | |

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–02 | Record ID: C"MOD" |

>**03**   Action
>
>   - C"Q" FUNCTION=QUIESCE
>
>   - C"N" FUNCTION=NORMAL
>
>**04–06**   Flag bytes from the CV4580
>
>**07**   Number of days left before NDREC is garbage collected
>
>**08**   Flag byte
>
>>   **Bit**   **Meaning**
>>   **11.. ....**
>>>   Node type from NDREC
>>>
>>>   **B′00′**   End node
>>>
>>>   **B′01′**   Network node
>>>
>>>   **B′10′**   Virtual node
>>
>>   **..1. ....**
>>>   Garbage collection indicator from NDREC
>>
>>   **...1 ....**
>>>   Adjacent node indicator from NDREC
>>
>>   **.... 1...**
>>>   DLUR end node indicator from NDREC
>
>**09**   Scope of the MODIFY TOPO command
>
>   - C"L" SCOPE=LOCAL
>
>   - C"N" SCOPE=NETWORK
>
>**0A–17**   0
>
>**18–1B**   Resource sequence number from CV4580
>
>**1C–1F**   First word of timestamp for trace entry

# NDREC TDU entry for topology database update

>**Entry:** **TDU**
>**Event:** Topology database update for a node
>**Modules Generating Entry:**
>>   ISTTRINP

This trace record is issued when a node record is created or updated when a TDU is received for a node and the information in the TDU about that node is more recent than the information already in the NDREC.

| 0 0 0 | 0 3 | 0 0 0 4 5 6 | 0 7 | 0 0 0 0 0 0 0 0 8 9 A B C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|---|
| TDU | A C T I O N | V4580 FLAGS | F L A G S | ADJACENT NODE SENDING TDU | CURRENT FRSN | LAST FRSN | V4580 RSN | TIME- STAMP |

**Byte (hex)**
>   **Contents**
>**00–02**   Record ID: C"TDU"
>**03**   Action
>
>   - C"C" Create node record
>
>   - C"U" Update node characteristics

- C"R" Node has changed node roles

04–06   Flag bytes from the CV4580
07      Flag byte

**Bit     Meaning**
**11.. ....**
        Node type from NDREC

    **B'00'**   End node

    **B'01'**   Network node

    **B'10'**   Virtual node

**..1. ....**
        Garbage collection indicator from NDREC
**...1 ....**
        Adjacent node indicator form NDREC
**.... 1...**
        DLUR end node indicator from NDREC

08–0F   CPNAME of the adjacent node sending the TDU. The CPNAME is not
        network qualified. This is the node that forwarded the TDU and is not
        necessarily the originator of the TDU information.
10–13   Current FRSN (flow reduction sequence number)
14–17   Last FRSN previously sent from the adjacent node that forwarded this
        TDU
18–1B   Resource sequence number from CV4580
1C–1F   First word of timestamp for trace entry

# NDREC TGU entry for TG update

**Entry:  TGU**
**Event:** Node created or updated because of a TG update signal
**Modules Generating Entry:**
        ISTTRTLT, ISTTRTTG

This trace record is issued when a node record is created or node information is
changed because of a TG update signal.

| 0 0 0 | 0 | 0 0 0 | 0 | 0 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 | 7 | 8 | 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B C | D E F |
| TGU | A C T I O N | V4580 FLAGS | N D T I M E | F L A G S | 0 | V4580 RSN | TIME- STAMP |

**Byte (hex)**
    **Contents**
00–02   Record ID: C"TGU"
03      Action

- C"C" Create node record

- C"U" Update node characteristics

- C"R" Node has changed node roles

04–06   Flag bytes from the CV4580
07      Number of days left before NDREC is garbage collected

**08** Flag byte

> **Bit** **Meaning**
> **11.. ....**
> > Node type from NDREC
> >
> > **B′00′** End node
> >
> > **B′01′** Network node
> >
> > **B′10′** Virtual node
>
> **..1. ....**
> > Garbage collection indicator from NDREC
> **...1 ....**
> > Adjacent node indicator from NDREC
> **.... 1...**
> > DLUR end node indicator from NDREC

**09–17** 0
**18–1B** Resource sequence number from CV4580
**1C–1F** First word of timestamp for trace entry

# NDREC TOP entry for topology data set create

**Entry:** TOP
**Event:** Node created from a checkpointed topology data set
**Modules Generating Entry:**
> ISTTRHND

This trace record is issued when a node record is created from a checkpointed topology data set.

| 0 0 0 | 0 | 0 0 0 | 0 | 0 | 0 0 0 0 0 0 0 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 5 6 | 7 | 8 | 9 A B C D E F | 0 1 2 3 | 4 5 6 7 | 8 9 A B | C D E F |
| TOP | A C T I O N | V4580 FLAGS | N D T I M E | F L A G S | 0 | TOPO CURRENT FRSN | TOPO LAST FRSN | V4580 RSN | TIME-STAMP |

**Byte (hex)**
> **Contents**
**00–02** Record ID: C"TOP"
**03** Action

> • C"C" Create node record

**04–06** Flag bytes from the CV4580
**07** Number of days left before NDREC is garbage collected
**08** Flag byte

> **Bit** **Meaning**
> **11.. ....**
> > Node type from TOPO data set
> >
> > **B′00′** End node
> >
> > **B′01′** Network node
> >
> > **B′10′** Virtual node

```
..1. ....
```
Adjacent node indicator from the TOPO data set

**09–0F**  0

**10–13**  Current FRSN (flow reduction sequence number) from the TOPO data set

**14–17**  Last FRSN received from this adjacent node from the TOPO data set

**18–1B**  Resource sequence number from CV4580 in the TOPO data set

**1C–1F**  First word of timestamp for trace entry

## The TGREC trace table

The TGREC trace table contains entries describing the creation and update of a TG record. It is located following the CV46 in a TGREC. A trace pointer in the TGREC points to it. The first 8 bytes of the TGREC trace table is the header, which contains the following information:

**Byte (hex)**
> **Contents**

**00–03**  The current RSN (resource sequence number) of the last update to this TGREC. The current RSN is used when recording TDUs (topology database updates) related to the TGREC. Because duplicate TDU information can be received from multiple adjacent nodes, the RSN in the TDU is compared to the current RSN in the TGREC trace. The TDU will only be recorded in the TGREC trace if the RSN in the TDU is greater than the current RSN in the TGREC trace, indicating that the information in the TDU is more recent than that in the TGREC.

**04–07**  Address of the current TGREC trace table entry. This is the last trace entry where data was recorded.

Following the header are five TGREC trace table entries. The information about the creation of the TGREC is recorded in the first TGREC trace entry. This entry remains unchanged for the life of the TGREC. Recording of trace events will continue in the remaining four trace table entries, with the trace wrapping back to the second entry after the fifth entry is completed.

## TGREC trace record descriptions

This topic contains the TGREC trace record descriptions. The entries are listed alphabetically by entry name.

### TGREC CPC entry for CP-CP session changes for a TG

**Entry:  CPC**
**Event:**  CP-CP session status changes in a TG record
**Modules Generating Entry:**
> ISTTRPCS

This trace record is issued for a TG record when the status of a CP-CP session using a local TG has changed.

## TGREC CPC

| 0 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| CPC | | A C T I O N | V 4 7 S T A T | V 4 6 F L G S | F L A G S | T G T I M E | F L A G S 2 | T G N | 0 | | | CW CGID | | | CL CGID | | | FRSN | | | V47 RSN | | | TIME- STAMP | | |

**Byte (hex)**

**Contents**

**00–02** Record ID: C"CPC"

**03** Action

- C"U" Update TG characteristics

**04** Status byte from the CV47

**05** Flag byte from the CV4680

**06** Flag byte

**Bit** **Meaning**

**111. ....**

TG type from TGREC

**B'000'** Endpoint TG

**B'001'** Intermediate routing TG

**B'010'** Interchange TG

**B'011'** Intersubnet TG

**B'100'** Branch Extender TG

**...1 ....**

Garbage collection indicator from TGREC

**.... 1...**

Adjacent node's HPRTT value from TGREC

**.... .11.**

Local node's HPR/HPRTT values from TGREC

**.... ...1**

TG carries conwinner CP-CP session indicator from TGREC

**07** Number of days left before TGREC is garbage collected

**08** Flag2 byte

**Bit** **Meaning**

**11.. ....**

CP-CP session status from CP status IPS

**B'01'** Pending active

**B'11'** Active

**B'00'** Inactive

**..11 ....**

CP-CP session type from CP status IPS

**B'01'** Conwinner

**B'00'** Conloser

**B'10'** Both conwinner and conloser

> **.... 1...**
>> Adjacent node type from CP status IPS
>>
>>> **B′0′** End node
>>>
>>> **B′1′** Network node
>
> **.... .1..**
>> Adjacent node's non-native indicator from CP status IPS
>>
>>> **B′0′** Native
>>>
>>> **B′1′** Non-native (can still have the same NETID)
>
> **.... ..1.**
>> Adjacent node's border node indicator from CP status IPS
>>
>>> **B′0′** Not a BN
>>>
>>> **B′1′** BN
>
> **.... ...1**
>> Adjacent node's border node supported indicator from CP status IPS
>>
>>> **B′0′** BN not supported
>>>
>>> **B′1′** BN supported

**09** TG number from CP status IPS (from RTP only)
**0A-0B** 0
**0C-0F** Conwinner CGID from CP status IPS
**10-13** Conloser CGID from CP status IPS
**14-17** Last FRSN received by adjacent node (from this host node) from CP status IPS
**18-1B** Resource sequence number from CV47
**1C-1F** First word of timestamp for trace entry

# TGREC MOD entry for MODIFY TOPO command

**Entry: MOD**
**Event:** MODIFY TOPO operator command
**Modules Generating Entry:**
> ISTTROMT

This trace record is issued when the MODIFY TOPO command with the FUNCTION=QUIESCE or FUNCTION=NORMAL operand is entered by an operator.

| 0 0 0 | 0 | 0 | 0 | 0 | 0 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 | 1 1 1 1 | 1 1 1 1 |
|---|---|---|---|---|---|---|---|---|
| 0 1 2 | 3 | 4 | 5 | 6 | 7 | 8 9 A B C D E F 0 1 2 3 4 5 6 7 | 8 9 A B | C D E F |
| MOD | A C T I O N | V 4 7 S T A T | V 4 6 F L G S | F L A G S | T G T I M E | S C O P E                        0 | V47 RSN | TIME- STAMP |

**Byte (hex)**
> **Contents**

**00–02** Record ID: C″MOD″
**03** Action

- C"Q" FUNCTION=QUIESCE

- C"N" FUNCTION=NORMAL

**04**    Status byte from the CV47

**05**    Flags byte from the CV4680

**06**    Flag byte

| Bit | Meaning |
|---|---|
| **111. ....** | TG type from TGREC |

    **B'000'**   Endpoint TG

    **B'001'**   Intermediate routing TG

    **B'010'**   Interchange TG

    **B'011'**   Intersubnet TG

    **B'100'**   Branch extender TG

**...1 ....**
    Garbage collection indicator from TGREC

**.... 1...**
    Adjacent node's HPRTT value from TGREC

**.... .11.**
    Local node's HPR/HPRTT values from TGREC

**07**    Number of days left before TGREC is garbage collected

**08**    Scope of the MODIFY TOPO command

- C"L" SCOPE =LOCAL

- C"N" SCOPE =NETWORK

**09–17**    0

**18–1B**    Resource sequence number from CV47

**1C–1F**    First word of timestamp for trace entry

## TGREC TDU entry for topology database update

**Entry:** TDU
**Event:** Topology database update for a TG
**Modules Generating Entry:**
    ISTTRIAT

This trace record is issued when a TG record is created or updated when a TDU is received for a TG and the information in the TDU about that TG is more recent than the information already in the TGREC.

| 0 0 0 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 8 0 9 A B C D E F | 1 0 1 1 2 3 | 1 4 5 6 7 | 1 8 9 A B | 1 C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| TDU | A C T I O N | V 4 7 S T A T | V 4 6 F L G S | F L A G S | T G T I M E | ADJACENT NODE SENDING TDU | CURRENT FRSN | LAST FRSN | V47 RSN | TIME-STAMP |

**Byte (hex)**
    **Contents**
**00–02**    Record ID: C"TDU"
**03**    Action

- C"C" Create TG record
- C"I" TG became inoperative
- C"O" TG became operational

**04** Status byte from the CV47

**05** Flag byte from the CV4680

**06** Flag byte

| **Bit** | **Meaning** |
|---|---|
| **111. ....** | |

TG type from TGREC

**B'000'** Endpoint TG

**B'001'** Intermediate routing TG

**B'010'** Interchange TG

**B'011'** Intersubnet TG

**B'100'** Branch extender TG

**...1 ....**

Garbage collection indicator from TGREC

**.... 1...**

Adjacent node's HPRTT value from TGREC

**.... .11.**

Local node's HPR/HPRTT values from TGREC

**07** Number of days left before TGREC is garbage collected

**08–0F** CPNAME of the adjacent node sending the TDU. The CPNAME is not network qualified. This is the node that forwarded the TDU and is not necessarily the originator of the TDU information.

**10–13** Current FRSN (flow reduction sequence number)

**14–17** Last FRSN previously sent from the adjacent node that forwarded this TDU

**18–1B** Resource sequence number from CV47

**1C–1F** First word of timestamp for trace entry

# TGREC TGU entry for TG update

**Entry: TGU**
**Event:** TG created or updated because of a TG update signal
**Modules Generating Entry:**
ISTTRTTG

This trace record is issued when a TG record is created or the TG status is changed because of a TG update signal.

| 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TGU | | | ACTION | V47STAT | V46FLGS | FLAGS | TGTIME | 0 | | | | | | | | | | | | TG UPDATE INFO | | | | V47 RSN | | | | TIME-STAMP | | | |

**Byte (hex)**
**Contents**

## TGREC TGU

**00–02**   Record ID: C"TGU"
**03**   Action

- C"C" Create TG record
- C"I" TG became inoperative
- C"O" TG became operational

**04**   Status byte from the CV47
**05**   Flag byte from the CV4680
**06**   Flag byte

| Bit | Meaning |
|-----|---------|
| **111. ....** | TG type from TGREC |

       **B′000′**   Endpoint TG

       **B′001′**   Intermediate routing TG

       **B′010′**   Interchange TG

       **B′011′**   Intersubnet TG

       **B′100′**   Branch extender TG

**...1 ....**
       Garbage collection indicator from TGREC
**.... 1...**
       Adjacent node's HPRTT value from TGREC
**.... .11.**
       Local node's HPR/HPRTT values from TGREC

**07**   Number of days left before TGREC is garbage collected
**08–13**   0
**14–17**   Information from the TG update signal

    **14**   TG type

       **X'01'**   Endpoint TG

       **X'02'**   Interchange TG

       **X'03'**   Intermediate routing TG

       **X'04'**   Intersubnet TG

       **X'05'**   Branch Extender TG

    **15**   TG status

       **X'01'**   Active

       **X'02'**   Inactive

    **16**   Partner node type

       **X'01'**   Network node

       **X'02'**   Virtual node

       **X'03'**   End node

    **17**   Direction of OP/INOP status

       **X'01'**   IN

       **X'02'**   OUT

       **X'03'**   IN/OUT

**18–1B**   Resource sequence number from CV47
**1C–1F**   First word of timestamp for trace entry

## TGREC TOP entry for topology data set create

**Entry: TOP**
**Event:** TG created from a checkpointed topology data set
**Modules Generating Entry:**
    ISTTRHTG

This trace record is issued when a TG record is created from a checkpointed topology data set.

| 0 0 0 | 0 3 | 0 4 | 0 5 | 0 6 | 0 7 | 0 0 0 0 0 0 0 0 8 9 A B C D E F | 1 1 1 1 0 1 2 3 | 1 1 1 1 4 5 6 7 | 1 1 1 1 8 9 A B | 1 1 1 1 C D E F |
|---|---|---|---|---|---|---|---|---|---|---|
| TOP | A C T I O N | V 4 7 S T A T | V 4 6 F L G S | F L A G S | T G T I M E | 0 | TOPO CURRENT FRSN | TOPO LAST FRSN | V47 RSN | TIME-STAMP |

**Byte (hex)**

    **Contents**

**00–02**  Record ID: C"TOP"

**03**     Action

       • C"C" Create TG record

**04**     Status byte from the CV47

**05**     Flags byte from the CV4680

**06**     Flag byte

       **Bit**    **Meaning**

       **111. ....**

            TG type from TGREC

            **B′000′**  Endpoint TG

            **B′001′**  Intermediate routing TG

            **B′010′**  Interchange TG

            **B′011′**  Intersubnet TG

            **B′100′**  Branch extender TG

       **...1 ....**

            Garbage collection indicator from TGREC

       **.... 1...**

            Adjacent node's HPRTT value from TGREC

       **.... .11.**

            Local node's HPR/HPRTT values from TGREC

**07**     Number of days left before TGREC is garbage collected

**08–0F**  0

**10–13**  Current FRSN (flow reduction sequence number) from the TOPO data set

**14–17**  Last FRSN received from the TOPO data set

**18–1B**  Resource sequence number from CV47 in the TOPO data set

**1C–1F**  First word of timestamp for trace entry

## The TRS common topology trace table

In the TRS (topology and routing services component of VTAM) common topology trace table, the deletion of NDRECs (node records) and TGRECs (TG records) are recorded.

The TRS common topology trace table is located in extended private storage. At TRS initialization, a 40 K buffer of storage is allocated for the TRS topology trace. When that buffer is filled with trace entries, another buffer is allocated to continue the trace. Buffer allocation continues until a maximum of 30 buffers is allocated for the TRS topology trace. When all buffers of the TRS topology trace are full of trace entries, the trace wraps back to the first entry on the first buffer of the trace table.

The TRS topology trace table is pointed to from the TRDAT. The following information about the TRS topology trace table is included in the TRDAT:

- A pointer to the first buffer of the TRS topology trace
- A pointer to the last buffer of the TRS topology trace
- A pointer to the current buffer of the TRS topology trace
- A pointer to the current TRS topology trace entry
- The number of trace records that have been used in the current buffer of the TRS topology trace
- The number of times the TRS topology trace table has wrapped
- The number of TRS topology trace table buffers allocated

Figure 6 on page 423 shows the TRS topology trace table format:
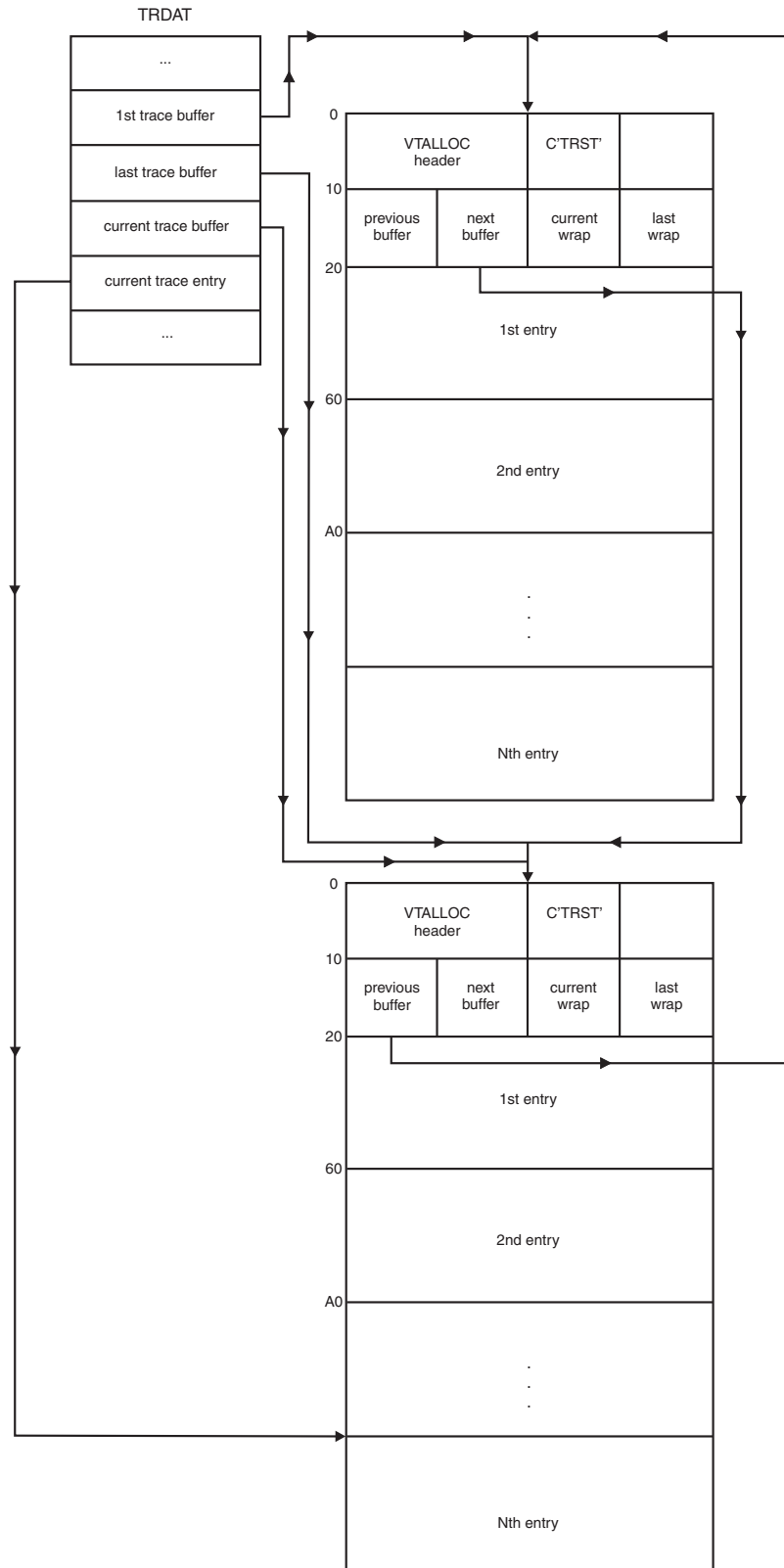
*Figure 6. TRS common topology trace table with two buffers allocated*

The first X'18' bytes of the allocated buffer of TRS topology trace table is the header, which contains the following information:

**Byte (hex)**

**Contents**

| | |
|---|---|
| 00–03 | TRS topology trace control block ID – C"TRTR" |
| 04–07 | 0 |
| 08–0B | Address of the previous buffer of the TRS topology trace. The previous buffer pointer is 0 for the first buffer of the trace. |
| 0C–0F | Address of the next buffer of the TRS topology trace. The next buffer pointer is 0 for the last buffer of the trace. |
| 10–13 | Timestamp (first word) when the trace most recently wrapped back to the beginning. The timestamp is present only on the first buffer of the TRS topology trace. |
| 14–17 | Prior timestamp (first word) when the trace wrapped back to the beginning. The timestamp is present only on the first buffer of the TRS topology trace. |

# TRS common topology trace record descriptions

This topic contains the TRS topology trace record descriptions. Each TRS topology trace entry is X'40' bytes long. The record ID in the first two characters of the record indicate the following:

- Byte 0 – Reason for the record deletion or TDU entry. See the individual record descriptions for reasons.
- Byte 1 – Type of entry

C"N"  – Node

C"T"  – TG

## TRS DELETE entry for NDREC deletion

**Event:** Node record deletion

This trace record is issued when a node record is deleted from the topology database.

| 00 | 01 | 02 03 04 | 05 | 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 | 17 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|
| DELETE | ID | V4580 FLAGS | FLAGS | NODE CPNAME | 0 |

| 20 21 22 23 24 25 26 27 | 28 | 29 | 2A | 2B | 2C 2D 2E 2F | 30 31 32 33 | 34 35 36 37 | 38 39 3A 3B | 3C 3D 3E 3F |
|---|---|---|---|---|---|---|---|---|---|
| 0 | CREATE | SOURCE | 0 | | CREATE RSN | DELETE RSN | CREATE TIME | DELETE TIME | NDREC ADDRESS |

**Byte (hex)**
> **Contents**

**00–01**

| Record ID | Calling module | Reason for node deletion |
|---|---|---|
| C"GN" | ISTTRGTM | Garbage collection |
| C"MN" | ISTTROMT | MODIFY TOPO,DELETE,SCOPE=LOCAL |
| C"SN" | ISTTRGTM | Garbage collection due to MODIFY TOPO,DELETE,SCOPE=NETWORK |
| C"TN" | ISTTRGTM | Garbage collection due to receipt of a TDU with the GCI bit on |

**02–04** Flag bytes from the CV4580

**05** Flag byte

> **Bit** **Meaning**
>
> **11.. ....**
> > Node type from NDREC
> >
> > **B′00′** End node
> >
> > **B′01′** Network node
> >
> > **B′10′** Virtual node
>
> **..1. ....**
> > Adjacent node indicator from NDREC
>
> **...1 ....**
> > Garbage collection indicator from NDREC
>
> **.... .1..**
> > DLUR end node indicator from NDREC
>
> **.... ..1.**
> > Scope of MODIFY TOPO
> >
> > **B′0′** Local
> >
> > **B′1′** Network

**06–16** Network qualified CPNAME of the node being deleted

**17–27** 0

**28–2A** Source of the node record creation

- C"TDU" Created due to TDU received from an adjacent node
- C"TGU" Created as the result of a TG update signal
- C"TOP" Created from a checkpointed data set at VTAM initialization with INITDB=TOPO

**2B** 0

**2C–2F** Resource sequence number at the time the node record was created

**30–33** Resource sequence number at the time the node record was deleted

**34–37** First word of timestamp when the node record was created

**38–3B** First word of timestamp when the node record was deleted

**3C–3F** Address of the node record being deleted

## TRS DELETE entry for TGREC deletion

**Event:** TG record deletion

This trace record is issued when a TG record is deleted from the topology database.

## TRS DELETE

| 00 01 | 02 | 03 | 04 | 05 | 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 | 17 18 19 1A 1B 1C 1D 1E 1F |
|---|---|---|---|---|---|---|
| DELETE ID | TGN | V47STAT | V46FLGS | FLAGS | ORIGIN CPNAME | DESTINATION CPNAME |

| 20 21 22 23 24 25 26 27 | 28 | 29 | 2A | 2B | 2C 2D 2E 2F | 30 31 32 33 | 34 35 36 37 | 38 39 3A 3B | 3C 3D 3E 3F |
|---|---|---|---|---|---|---|---|---|---|
| DESTINATION CPNAME (CONT) | CREATE | SOURCE | 0 | CREATE RSN | DELETE RSN | CREATE TIME | DELETE TIME | TGREC ADDRESS |

**Byte (hex)**
    **Contents**
**00–01**

| Record ID | Calling module | Reason TG deletion |
|---|---|---|
| C"BT" | ISTTRPCS | Node role change from EN to BN |
| C"CT" | ISTTRINP | Node role change via a TDU |
| C"DT" | ISTTRPSS | Deactivation of CPSERVR |
| C"FT" | ISTTRIAT | Zero FRSN received on a TDU from an EN |
| C"GT" | ISTTRGTM | Garbage collection |
| C"HT" | ISTTRTTG | Node role change by way of a TG update |
| C"MT" | ISTTROMT | MODIFY TOPO,DELETE,SCOPE=LOCAL |
| C"NT" | ISTTRGNR | Deleted as the result of a node deletion |
| C"RT" | ISTTRGTR | Reverse TG deletion |
| C"ST" | ISTTRGTM | Garbage collection due to MODIFY TOPO,DELETE,SCOPE=NETWORK |
| C"TT" | ISTTRGTM | Garbage collection due to receipt of a TDU with the GCI bit on |

**02**    TG number from the CV4680
**03**    Status byte from the CV47
**04**    Flag byte from the CV4680
**05**    Flag byte

**Bit    Meaning**
**111. ....**
            TG type from TGREC

            **B'000'**   Endpoint TG

            **B'001'**   Intermediate routing TG

            **B'010'**   Interchange TG

            **B'011'**   Intersubnet TG

        **B′100′**   Branch extender TG

**...1 ....**
        Garbage collection indicator from TGREC

**.... ..1.**
        Scope of MODIFY TOPO

        **B′0′**     Local

        **B′1′**     Network

| | |
|---|---|
| **06–16** | Network qualified CPNAME of the origin node of the TG being deleted |
| **17–27** | Network qualified CPNAME of the destination node of the TG being deleted |
| **28–2A** | Source of the TG record creation |

    • C"TDU" Created due to TDU received from an adjacent node

    • C"TGU" Created as the result of a TG update signal

    • C"TOP" Created from a checkpointed data set at VTAM initialization with INITDB=TOPO

| | |
|---|---|
| **2B** | 0 |
| **2C–2F** | Resource sequence number at the time the TG record was created |
| **30–33** | Resource sequence number at the time the TG record was deleted |
| **34–37** | First word of timestamp when the TG record was created |
| **38–3B** | First word of timestamp when the TG record was deleted |
| **3C–3F** | Address of the TG record being deleted |

**TRS DELETE**

# Appendix C. First Failure Support Technology (FFST) probes

This appendix contains the following topics:
- "FFST probe index"
- "FFST probe information" on page 430
- "FFST probe naming conventions" on page 430
- "FFST probe descriptions" on page 431

See "First Failure Support Technology (FFST) for VTAM" on page 1 for additional information.

## FFST probe index

The following table provides an index of FFST probes in alphanumerical order by probe name:

*Table 8. FFST probe index*

| Probe name | Component | See |
|---|---|---|
| ISTALCxx | APPN Over Logical Link Control | "APPN over logical link control probes" on page 431 |
| ISTATCxx | APPN TG Management | "APPN TG management" on page 432 |
| ISTAUCxx | Enterprise Extender | "Enterprise Extender probes" on page 449 |
| ISTCMCxx | Common Management Information Protocol (CMIP) Services | "Common Management Information Protocol (CMIP) services probes" on page 432 |
| ISTCOCxx | Control Operator | "Control operator probes" on page 434 |
| ISTCSCxx | Configuration Services | "Configuration services probes" on page 433 |
| ISTDLCxx | Dependent LU Server | "Dependent LU server probes" on page 441 |
| ISTDRCxx | Directory Services | "Directory services probes" on page 442 |
| ISTFSCxx | Coupling Facility Services | "Coupling facility services probes" on page 435 |
| ISTIUTxx | Logical Link Control | "Logical link control" on page 450 |
| ISTMTCxx | Management Services Transport | "Management services transport probes" on page 451 |
| ISTNACxx | Network Resource Management | "Network resource management probes" on page 452 |
| ISTNSCxx | LU Network Services | "LU network services probes" on page 451 |
| ISTORCxx | Storage Management | "Storage management probes" on page 463 |
| ISTPSCxx | Presentation Services | "Presentation services probes" on page 453 |
| ISTRACxx | Trace Services | "Trace services probes" on page 464 |
| ISTRMCxx | LU Resource Manager | "LU resource manager probes" on page 451 |
| ISTRVMxx | Recovery Manager | "Recovery manager (RV) probes" on page 453 |
| ISTSCCxx | Session Services CP-CP | "Session services CP-CP probes" on page 454 |

*Table 8. FFST probe index  (continued)*

| Probe name | Component | See |
|---|---|---|
| ISTSLCxx | Session Services LU-LU | "Session services LU-LU probes" on page 456 |
| ISTSSCxx | Session Services | "Session services probes" on page 454 |
| ISTTACxx | VTAM Topology Agent | "VTAM topology agent probes" on page 466 |
| ISTTCCxx | TC-DLC (Transmission Control-Data Link Control) | "Transmission control — Data link control (TC-DLC) probes" on page 464 |
| ISTTRCxx | Topology and Routing Services | "Topology and routing services probes" on page 463 |
| ISTTSCxx | Transmission Subsystem | "Transmission subsystem probes" on page 465 |

# FFST probe information

When a VTAM FFST probe is triggered, an unexpected condition has occurred in the network. The process that received the condition might not complete normally. The VTAM program will attempt to recover from the unexpected condition and will continue processing subsequent requests. Recovery might not be possible for some system conditions, and subsequent requests might fail, terminals might hang, and other abnormal conditions might occur.

Dump data is collected to assist in identifying the source of the problem. The processing element is freed and processing continues. If the probe triggers multiple times you might need to halt and restart VTAM.

Contact the appropriate IBM Support Center and supply the service representative with the console listing that is written at the time of the error and the dump data produced by the probe.

For information on using FFST dumps, see "First Failure Support Technology (FFST) for VTAM" on page 1.

# FFST probe naming conventions

The following table describes the naming convention for the FFST probe name. **ISTRAC01** is used as an example for VTAM.

*Table 9. FFST probe naming conventions*

| Characters | Example | Description |
|---|---|---|
| 1,2,3 | IST | These characters represent the product identifier. For VTAM probes, these characters are IST. |
| 4, 5 | RA | These characters represent the VTAM component identifier. In this example, RA is the component identifier for the Trace Services component. |
| 6 | C | For VTAM probes, this character is usually C. |
| 7, 8 | 01 | These characters represent the probe identification number. This number is not duplicated. |

# FFST probe descriptions

This information includes a table for each component that contains FFST probe instructions. The components are in alphabetical order, and the probes for each component are in alphanumerical order by probe name. "FFST probe index" on page 429 provides an index of FFST probes in alphanumerical order by probe name. Each table in this information shows the probe name, the module that issues it, and whether the probe creates a full or minidump when triggered.

"FFST probe index" on page 429 provides an index of FFST probes in alphanumerical order by probe name.

For information on using FFST dumps, see "First Failure Support Technology (FFST) for VTAM" on page 1.

## APPN over logical link control probes

Table 10. FFST probes for APPN over logical link

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTALC01 | ISTALCAL | The APPN over logical link control (ALLC) line index control block OSLIN indicates that ALNCB line entries are available, but none can be found. The maximum number of ATM native ALNCB line entries is 65535. | Mini |
| ISTALC02 | ISTALCPE | An unusable primitive was received by ISTALCPE. | Mini |
| ISTALC03 | ISTALCPF | An unusable work element was queued to the ALLC port finite state machine (FSM) processor ISTALCPF. | Mini |
| ISTALC04 | ISTALCPF | The work in progress field (ALPOR_WIP) in the ALLC port control block ALPOR has not been set and should have been set during DACTLINK processing. | Mini |
| ISTALC05 | ISTALCPF | Disable confirmation was received and the filter has never gone inactive. | Mini |
| ISTALC06 | ISTALCPF | The work element address (WEA) field in the request processing header control block RPH does not point to an ALLC parameter list (ALCPL). | Mini |
| ISTALC07 | ISTALCLF | An unusable work element was received by ISTALCLF. | Mini |
| ISTALC08 | ISTALCIF | An unusable work element was received by ISTALCIF. | Mini |
| ISTALC09 | ISTALCOF | An unusable work element was received by ISTALCOF. | Mini |
| ISTALC10 | ISTALCCF | An unusable work element was received by ISTALCCF. | Mini |
| ISTALC11 | ISTALCOC | An unusable AAL signal was received by ISTALCOC from the AAL FSM handler ISTALCAF. | Mini |

*Table 10. FFST probes for APPN over logical link (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTALC12 | ISTALCIC | An unusable AAL signal was received by ISTALCIC from the AAL FSM handler ISTALCAF. | Mini |
| ISTALC13 | ISTALCOL | An unusable XID signal was received by ISTALCOL from the XID FSM handler ISTALCXF. | Mini |
| ISTALC14 | ISTALCIL | An unusable XID signal was received by ISTALCIL from the XID FSM handler ISTALCXF. | Mini |

## APPN TG management

*Table 11. FFST probes for APPN TG management*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTATC01 | ISTATCUP | Total connection count corrupted | FULL |

## Common Management Information Protocol (CMIP) services probes

*Table 12. FFST probes for CMIP services (ISTCMCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTCMC01 | ISTCMAFR | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC02 | ISTCMCRT | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC03 | ISTCMDCR | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC04 | ISTCMIOS | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC05 | ISTCMMBR | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC06 | ISTCMRPR | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC07 | ISTCMSSR | CMIP services was dispatched with a work element type that it does not support or does not support in the current state. | Full |
| ISTCMC08 | ACYCMLNC | CMIP services attempted to send an internal CMIP services signal to a destination that is not valid. | Full |

*Table 12. FFST probes for CMIP services (ISTCMCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTCMC09 | ACYCMPTC | CMIP services called the PTIO_OPEN function and sent parameters that are not valid. | Full |

## Configuration services probes

*Table 13. FFST probes for configuration services (ISTCSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTCSC01 | ISTACCQ3 | This probe detects control vector (or subvector) lengths that are not valid on request contact (REQCONT) RUs. It produces information about the failing request unit processing element (RUPE), request/response unit (RU), and variable work area (VWA). If available, the resource definition table entry (RDTE) for the LINE and PU is also reported. | Mini |
| ISTCSC02 | ISTCSCRC | Resource registration failure was detected. | Mini |
| ISTCSC03 | ISTCSCRE | Resource registration failure was detected. | Mini |
| ISTCSC04 | ISTCSCRF | Resource registration failure was detected. | Mini |
| ISTCSC05 | ISTCSCRJ | Resource registration failure was detected. | Mini |
| ISTCSC06 | ISTCSCRK | Resource registration failure was detected. | Mini |
| ISTCSC07 | ISTCSCRH | Resource registration failure was detected. | Mini |
| ISTCSC09 | ISTDECQE | The resource definition table entry (RDTE) for the dependent LU requester CDRSC was not found. | Full |
| ISTCSC10 | ISTDECQD | The resource definition table entry (RDTE) for the dependent LU requester CDRSC was not found. | Full |
| ISTCSC11 | ISTDECP3 | The resource definition table entry (RDTE) for the dependent LU requester CDRSC was not found. | Full |
| ISTCSC12 | ISTDECP4 | The resource definition table entry (RDTE) for the dependent LU requester CDRSC was not found. | Full |
| ISTCSC13 | ISTDECQ4 | A protocol violation was detected. | Full |
| ISTCSC14 | ISTDECQ4 | A topology database update (TDU) error was detected. | Full |
| ISTCSC15 | ISTACCQE | An attempt to add the resource definition table entry (RDTE) for the DLUS-supported LU failed because the network address is already in use or was not properly freed by a previous user. Related pointer values are provided in the secondary symptom string. | Full |
| ISTCSC16 | ISTDECLL | The GETRDTE for the higher-level resource failed. | Full |
| ISTCSC18 | ISTACCQ3 | REQCONT failure RU is not valid. | Mini |

## Control operator probes

*Table 14. FFST probes for control operator (ISTCOCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTCOC01 | ISTCOCDL | A suspended log mode is not found.<br><br>Check the mode the application program specifies on the APPCCMD CONTROL=OPRCNTRL, QUALIFY=RESTORE macroinstruction.<br><br>You can use the following VTAM dump functions to display advanced program-to-program communication (APPC)-related information:<br>• APPLCONV<br>• APPLMODE<br>• APPMODAL<br>• PARTNRLU<br><br>For information about these functions, see the information about using VTAM dump analysis tools in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*. | Full |
| ISTCOC02 | ISTCOCSC | A suspended log mode is not found.<br><br>Check the mode the application program specifies on the APPCCMD CONTROL=OPRCNTRL, QUALIFY=RESTORE macroinstruction.<br><br>You can use the following VTAM dump functions to display advanced program-to-program communication (APPC)-related information:<br>• APPLCONV<br>• APPLMODE<br>• APPMODAL<br>• PARTNRLU<br><br>For information about these functions, see the information about using VTAM dump analysis tools in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*. | Full |

*Table 14. FFST probes for control operator (ISTCOCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTCOC03 | ISTCOCVR | A suspended log mode is not found.<br><br>Check the mode the application program specifies on the APPCCMD CONTROL=OPRCNTRL, QUALIFY=RESTORE macroinstruction.<br><br>You can use the following VTAM dump functions to display advanced program-to-program communication (APPC)-related information:<br>• APPLCONV<br>• APPLMODE<br>• APPMODAL<br>• PARTNRLU<br><br>For information about these functions, see the information about using VTAM dump analysis tools in *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures*. | Full |

# Coupling facility services probes

*Table 15. FFST probes for coupling facility services (ISTFSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC00 | ISTFSUMT | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to move an entry to the end of a list. | Mini |
| ISTFSC01 | ISTFSNRT | A work element that was not recognized by coupling facility services (CFS) was queued to the CFS connection services PAB. | Mini |
| ISTFSC02 | ISTFSNRT | A work element was recognized by coupling facility services (CFS), but it contained incorrect information. | Mini |
| ISTFSC03 | ISTFSNCF | An unexpected input was received for the current state of the connection finite state machine. | Mini |
| ISTFSC04 | ISTFSNST | An unexpected return code *xxxx* and reason code *yyyy* were returned for the IXLDISC macro. | Mini |
| ISTFSC05 | ISTFSURB | An unexpected return code *xxxx* and reason code *yyyy* were returned for the IXLREBLD macro. | Mini |
| ISTFSC06 | ISTFSURS | An unexpected return code *xxxx* and reason code *yyyy* were returned for the IXLEERSP macro. | Mini |

*Table 15. FFST probes for coupling facility services (ISTFSCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC07 | ISTFSUUS | An unexpected return code *xxxx* and reason code *yyyy* were returned for the IXLUSYNC macro. | Mini |
| ISTFSC08 | ISTFSUFC | An unexpected return code *xxxx* and reason code *yyyy* were returned for the IXLFORCE macro. | Mini |
| ISTFSC09 | ISTFSNCN | An unexpected return code *xxxx* and reason code *yyyy* were returned for the IXLCONN macro. | Mini |
| ISTFSC0A | ISTFSUDC | VTAM was connected to an unexpected structure type. | Mini |
| ISTFSC0B | ISTFSUDC | The connection name returned in the IXLCONN answer area was not the name specified on IXLCONN. | Mini |
| ISTFSC0C | ISTFSGLR | VTAM was not able to find a local generic mapping while rebuilding the coupling facility structure with local data. | Mini |
| ISTFSC0D | ISTFSDRT | A work element that was not recognized was received by a Sysplex Wide Security Associations structure object PAB. | Mini |
| ISTFSC0E | ISTFSGG1 | VTAM made a counting error associated with a generic resource. | Mini |
| ISTFSC0F | ISTFSUUS | Unexpected input was received for current state of the user sync point finite state machine (FSM). | Mini |
| ISTFSC10 | ISTFSCA3 | VTAM was unable to invoke the IWMGRREG macroinstruction for registering a generic name to the work load manager. | Mini |
| ISTFSC11 | ISTFSUDC | VTAM has disconnected from the generic resources coupling facility structure because a timed process did not complete within the given time. | Full |
| ISTFSC12 | ISTFSUXJ | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXCJOIN when VTAM attempted to join an XCF group. | Mini |
| ISTFSC13 | ISTFSUAL | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLALTER when VTAM attempted to alter coupling facility structure attributes. | Mini |

*Table 15. FFST probes for coupling facility services (ISTFSCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC14 | ISTFSLML | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLLIST when VTAM attempted to establish list monitoring. | Mini |
| ISTFSC15 | ISTFSNPG | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLPURGE when VTAM attempted to purge outstanding structure access request. | Mini |
| ISTFSC16 | ISTFSLRM | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLLIST when VTAM attempted to read multiple entries from a list. | Mini |
| ISTFSC17 | ISTFSLDM | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLLIST when VTAM attempted to delete multiple entries. | Mini |
| ISTFSC18 | ISTFSLWC | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLLIST when VTAM attempted to update the list control area. | Mini |
| ISTFSC19 | ISTFSLVC | An unexpected return code *xxxx* was returned from IXLVECTR when VTAM attempted to modify or test the list notification vector. | Mini |
| ISTFSC1A | ISTFSURL | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to read a list entry. | Mini |
| ISTFSC1B | ISTFSUUL | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to update a list entry. | Mini |
| ISTFSC1C | ISTFSUCL | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to create a list entry. | Mini |
| ISTFSC1D | ISTFSUHT | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to read and move a list entry. | Mini |
| ISTFSC1E | ISTFSKRT | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to set a lock entry. | Mini |

## FFST probes

*Table 15. FFST probes for coupling facility services (ISTFSCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC1F | ISTFSKRT | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to release a lock entry. | Mini |
| ISTFSC20 | ISTFSKRT | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to release a lock entry for a failed connector. | Mini |
| ISTFSC21 | ISTFSCFG | VTAM found a name defined as both a generic resource name and a USERVAR. | Mini |
| ISTFSC22 | ISTFSDRT | A work element was recognized by a Sysplex Wide Security Associations structure object PAB, but it contained incorrect data. | Mini |
| ISTFSC23 | ISTFSUDL | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to delete a list entry. | Mini |
| ISTFSC24 | ISTFSUEN | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to read list control information. | Mini |
| ISTFSC25 | ISTFSURH | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to read a list entry from the head of a list. | Mini |
| ISTFSC26 | ISTFSUWT | An unexpected return code *xxxx* and reason code *yyyy* were returned for IXLLIST when VTAM attempted to write a list entry to the end of a list. | Mini |
| ISTFSC27 | ISTFSLDE | An unexpected return code *xxxx* and reason code *yyyy* were returned from IXLLIST when VTAM attempted to delete a list of entries. | Mini |
| ISTFSC28 | ISTFSUQS | An unexpected return code and reason code were returned from IXCQUERY when VTAM attempted to obtain sysplex related information. | Mini |
| ISTFSC29 | ISTFSPDN | A pointer to a buffer containing NLP entry IDs for an IXLLIST DELETE_ENTRYLIST request that was being retried was zero. NLP entries may not be deleted from a multinode persistent session coupling facility structure. | Full |

*Table 15. FFST probes for coupling facility services (ISTFSCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC2A | | A coupling facility user deregistered but did not provide an area to pass back a work element when one existed. The storage for the work element is not freed. | Mini |
| ISTFSC2B | | A coupling facility user attempted to deregister, but provided a user token that was not valid. | Mini |
| ISTFSC2C | | A coupling facility user indicated a process was complete, but provided a user token that was not valid. | Mini |
| ISTFSC2D | ISTFSGRO | When attempting to alter the entry-to-element ratio for the generic resource structure, the new ratio was not valid. Either the entry portion was zero, the element portion was zero, or the entry portion divided by the element portion was greater than the maximum number of data elements. | Mini |
| ISTFSC2E | ISTFSLRO | When attempting to alter the entry-to-element ratio for a VTAM list structure, the new ratio was not valid. Either the entry portion was zero, the element portion was zero, or the entry portion divided by the element portion was greater than the maximum number of data elements. | Mini |
| ISTFSC30 | ISTFSPDN | An unexpected return code of X'0008' and reason code of X'0082B' were received for IXLLIST while attempting to delete NLP entries in the multinode persistent coupling facility structure. A start or stop index was given that was not valid. | Mini |
| ISTFSC31 | ISTFSPDN | An unexpected return code *xxxx* and reason code *yyyy* were received for IXLLIST while attempting to delete NLP entries in the multinode persistent coupling facility structure. NLP entries may not be deleted. | Mini |
| ISTFSC32 | ISTFSPRT | A work element that was not recognized was received by a multinode persistent session structure object PAB. | Mini |

# FFST probes

*Table 15. FFST probes for coupling facility services (ISTFSCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC33 | ISTFSPRT | A work element was recognized by a multinode persistent session structure object PAB, but it contained incorrect data. | Mini |
| ISTFSC34 | ISTFSPRV | An unexpected return code and reason code were returned from IXLLIST while attempting to read in a multinode persistent session (MNPS) application program's data during a MNPS recovery. Sessions will not be recovered. | Mini |
| ISTFSC35 | ISTFSLRL | An unexpected return code and reason code were returned from IXLLIST when attempting to read multiple entries from a list. | Mini |
| ISTFSC36 | ISTFSPDD | An unexpected return code and reason code were returned from IXLLIST when attempting to delete data associated with a multinode persistent session application program. The state of the application program will remain in CLEANUP and may be cleaned up by another VTAM. | Mini |
| ISTFSC37 | ISTFSDDD | An unexpected return code and reason code were returned from IXLLIST when attempting to delete data associated with a TCP/IP stack using Sysplex Wide Security Associations. | Mini |
| ISTFSC38 | ISTFSDRD | An unexpected return code and reason code were returned from IXLLIST when attempting to read data associated with a TCP/IP stack using Sysplex Wide Security Associations. | Mini |
| ISTFSC39 | ISTFSVRT | A work element that was not recognized was received by a Sysplexports structure object PAB. | Mini |
| ISTFSC3A | ISTFSVRT | A work element was recognized by a Sysplexports structure object PAB, but it contained incorrect data. | Mini |
| ISTFSC3C | ISTFSLLO | An unexpected return code and reason code were returned from IXLLIST when attempting to lock data associated with a structure object. | Mini |
| ISTFSC3D | ISTFSVDD | An unexpected return code and reason code were returned from IXLLIST when attempting to delete data associated with a TCP/IP stack using Sysplexports. | Mini |

*Table 15. FFST probes for coupling facility services (ISTFSCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTFSC3E | ISTFSCFG | Data that was not valid was returned by IXLLIST when attempting to read the generic resource mapping list of the generic resource structure. | Full |

## Dependent LU server probes

*Table 16. FFST probes for dependent LU server (ISTDLCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDLC01 | ISTDLCRD | Element representing the dependent LU requester could not be found | Mini |
| ISTDLC02 | ISTDLCRD | Element representing the PU could not be found. | Mini |
| ISTDLC03 | ISTDLCDP | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC04 | ISTDLCDR | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC07 | ISTDLCAL | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC08 | ISTDLCAL | Element representing the PU could not be found. | Full |
| ISTDLC09 | ISTDLCAS | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC10 | ISTDLCRT | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC11 | ISTDLCRT | Unrecognized signal; received signal is not supported by dependent LU server. | Mini |
| ISTDLC13 | ISTDLCSI | During processing, if a CP_SVR_SESS_STAT(inactive) signal is sent by the LRM component when CP-SVR pipe is deactivated, the combination of the contention winner and contention loser finite state machine states was found to be not valid. As a result, the final cleanup associated with the pipe deactivation could not be performed. | Full |
| ISTDLC20 | ISTDLCRP | Element representing the PU could not be found. | Mini |
| ISTDLC21 | ISTDLCAP | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC24 | ISTDLCTD | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC26 | ISTDLCDL | Element representing the dependent LU requester could not be found. | Mini |
| ISTDLC27 | ISTDLCDL | Element representing the PU could not be found. | Mini |
| ISTDLC28 | ISTDLCDL | Element representing DLUS-supported LU could not be found. | Mini |

*Table 16. FFST probes for dependent LU server (ISTDLCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDLC29 | ISTDLCSA | DLR FSM not found | Mini |
| ISTDLC30 | ISTDLCAQ | DLR PU FSM not expecting ACTPU REQ | Full |
| ISTDLC31 | ISTDLCAQ | DLR element not freed after inactive | Mini |
| ISTDLC33 | ISTDLCED | Element representing the dependent LU requester could not be found. | Mini |

## Directory services probes

*Table 17. FFST probes for directory services (ISTDRCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC00 | ISTDRCRT | An unrecognized request element was received. | Full |
| ISTDRC01 | ISTDRCRT | An unrecognized interprocess signal was received. | Full |
| ISTDRC02 | ISTDROBO | An unrecognized function code was received. | Full |
| ISTDRC04 | ISTDROFO | A build of a found CV80 is required but is not present. | Full |
| ISTDRC05 | ISTDROFO | A build of a found CV3C is required but is not present. | Full |
| ISTDRC06 | ISTDROFO | A build of a found CV3C is required but is not present. | Full |
| ISTDRC07 | ISTDROFO | A build of a found CV3D is required but is not present. | Full |
| ISTDRC08 | ISTDROFO | A build of a found CV3E is required but is not present. | Full |
| ISTDRC09 | ISTDROFO | A build of a found CV40 is required but is not present. | Full |
| ISTDRC0A | ISTDROLM | A build of a locate GDS variable is required but is not present. | Full |
| ISTDRC0B | ISTDROLM | A build of a locate GDS variable is required but is not present. | Full |
| ISTDRC0C | ISTDROLM | A build of a found GDS variable is required but is not present. | Full |
| ISTDRC0D | ISTDROLM | A build of a register GDS variable is required but is not present. | Full |
| ISTDRC0E | ISTDROLM | A build of a CDINIT GDS variable is required but is not present. | Full |
| ISTDRC0F | ISTDROLM | A build of an IOCD GDS variable is required but is not present. | Full |
| ISTDRC11 | ISTDRO0E | A build of a CV0E failed because the requested name was not recognized. | Full |
| ISTDRC12 | ISTDRO60 | A build of a CV60 failed because the requested name was not recognized. | Full |
| ISTDRC13 | ISTDROCG | The search type that was requested was not recognized. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC14 | ISTDROLM | A build of a notify GDS variable is required but is not present. | Full |
| ISTDRC15 | ISTDRDUE | An unknown entry type was received. | Full |
| ISTDRC16 | ISTDRDDD | Either an unknown resource or no resource was found for a resource delete request. | Full |
| ISTDRC17 | ISTDRDDD | The associated network ID information was not found for a resource delete request. | Full |
| ISTDRC18 | ISTDRDDD | A release of storage associated with a resource failed to complete. | Full |
| ISTDRC19 | ISTDRDDD | A release of storage associated with a resource failed to complete. | Full |
| ISTDRC1A | ISTDRDDD | A release of storage associated with a resource failed to complete. | Full |
| ISTDRC1B | ISTDRDUD | An update directory request was received, but no list elements were found. | Full |
| ISTDRC1C | ISTDRDCD | A release of storage associated with a resource failed to complete. | Full |
| ISTDRC1D | ISTDRDCD | A release of storage associated with a CACHE_DATA interprocess signal failed to complete. | Full |
| ISTDRC1E | ISTDRDCD | A CACHE_DATA interprocess signal failed because the requested name was not recognized. | Full |
| ISTDRC1F | ISTDRAPC | A CP_STATUS interprocess signal failed because the requested name was not recognized. | Full |
| ISTDRC20 | ISTDRAUA | Cannot remove an adjacent control point from the list. | Full |
| ISTDRC21 | ISTDRAUA | Cannot add an adjacent control point to the list. | Full |
| ISTDRC22 | ISTDRAUA | ISTDRAUA received an action code indicating that an unexpected condition has occurred. | Full |
| ISTDRC23 | ISTDRAUA | ISTDRAUA was unable to process an unrecognized action code. | Full |
| ISTDRC24 | ISTDREDS | An unexpected interprocess signal was received. | Full |
| ISTDRC25 | ISTDRFCH | An unrecognized interprocess signal was received for a node role. | Full |
| ISTDRC26 | ISTDRFLE | The sender of a request cannot be identified. | Full |
| ISTDRC27 | ISTDRFNV | An unknown verify indicator value was received. | Full |
| ISTDRC28 | ISTDRFSI | An unrecognized CV35 vector was received from a VTAM component. | Full |
| ISTDRC29 | ISTDRF35 | VTAM detected a software error while processing a CV35 vector. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC2A | ISTDRFLE | A protocol error was detected while processing an interprocess signal from a VTAM component. | Full |
| ISTDRC2B | ISTDRFNV | An unrecognized name was received from a VTAM component. | Full |
| ISTDRC2C | ISTDRFSI | A CV35 vector was received while the keep indicator was set. | Full |
| ISTDRC2D | ISTDRFPQ | An unrecognized network-qualified name was received. | Full |
| ISTDRC2E | ISTDRFPY | An unrecognized network-qualified name was received. | Full |
| ISTDRC2F | ISTDROLO | A build of a Locate CV80 is required, but none is present. | Full |
| ISTDRC30 | ISTDROLO | A build of a Locate CV60 is required, but none is present. | Full |
| ISTDRC31 | ISTDROLO | A build of a Locate CV81 is required, but none is present. | Full |
| ISTDRC32 | ISTDROLO | A build of a Locate CV2B is required, but none is present. | Full |
| ISTDRC33 | ISTDROLO | A build of a Locate CV0E is required, but none is present. | Full |
| ISTDRC34 | ISTDROLO | A build of a Locate CV35 is required, but none is present. | Full |
| ISTDRC35 | ISTDROFI | A build of a Find CV80 is required, but none is present. | Full |
| ISTDRC36 | ISTDROFI | A build of a Find CV3C is required, but none is present. | Full |
| ISTDRC37 | ISTDROFI | A build of a Find CV3C is required, but none is present. | Full |
| ISTDRC38 | ISTDROFI | A build of a Find CV3D is required, but none is present. | Full |
| ISTDRC39 | ISTDROFI | A build of a Find CV3E is required, but none is present. | Full |
| ISTDRC3A | ISTDROFI | A build of a Find CV81 is required, but none is present. | Full |
| ISTDRC3B | ISTDROFI | A build of a Find CV81 is required, but none is present. | Full |
| ISTDRC3C | ISTDROFI | A build of a Find CV82 is required, but none is present. | Full |
| ISTDRC3D | ISTDRMCH | An unrecognized input was detected. | Full |
| ISTDRC3E | ISTDRMCS | An unrecognized interprocess signal was received. | Full |
| ISTDRC3F | ISTDRMCS | An unrecognized state was detected. | Full |
| ISTDRC40 | ISTDRMDO | An unrecognized action was requested. | Full |
| ISTDRC41 | ISTDRMDO | An unexpected interprocess signal was received. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC42 | ISTDRMDO | Unexpected elements were found in a locate search. | Full |
| ISTDRC43 | ISTDRMDO | An unexpected interprocess signal was received. | Full |
| ISTDRC44 | ISTDRMDO | An unrecognized state was detected. | Full |
| ISTDRC45 | ISTDRMDO | An unexpected return code was received. | Full |
| ISTDRC46 | ISTDRMDO | A list requiring at least one element was empty. | Full |
| ISTDRC47 | ISTDRMFN | An unexpected interprocess signal was received. | Full |
| ISTDRC48 | ISTDRMFN | An unrecognized state was detected. | Full |
| ISTDRC49 | ISTDRMNV | An unexpected return code was received. | Full |
| ISTDRC4A | ISTDRMNV | An unexpected return code was received. | Full |
| ISTDRC4B | ISTDRMNV | An unexpected interprocess signal was received. | Full |
| ISTDRC4C | ISTDRMSA | An unexpected interprocess signal was received. | Full |
| ISTDRC4D | ISTDRMSA | An unrecognized state was detected. | Full |
| ISTDRC4E | ISTDRMTD | An unexpected interprocess signal was received. | Full |
| ISTDRC4F | ISTDRMTD | An unrecognized state was detected. | Full |
| ISTDRC50 | ISTDRRSD | Unrecognized inputs for state were detected. | Full |
| ISTDRC51 | ISTDRRSR | Unrecognized inputs for state were detected. | Full |
| ISTDRC52 | ISTDRRTP | A work element was received when none was expected. | Full |
| ISTDRC53 | ISTDRSBR | ISTDRSBR was unable to add a locate control block to the list. | Full |
| ISTDRC54 | ISTDRSNB | An unrecognized state was detected. | Full |
| ISTDRC55 | ISTDRSCH | An unexpected interprocess signal was received. | Full |
| ISTDRC56 | ISTDRSDB | An unexpected interprocess signal was received. | Full |
| ISTDRC57 | ISTDRSDB | An unrecognized state was detected. | Full |
| ISTDRC58 | ISTDRSDS | An unexpected return code was received. | Full |
| ISTDRC59 | ISTDRSDS | An unexpected return code was received. | Full |
| ISTDRC5A | ISTDRSDS | An unexpected return code was received. | Full |
| ISTDRC5B | ISTDRSDS | An unexpected return code was received. | Full |
| ISTDRC5C | ISTDRSDS | An unexpected return code was received. | Full |
| ISTDRC5D | ISTDRSDS | An unexpected interprocess signal was received. | Full |
| ISTDRC5E | ISTDRSDS | An unrecognized state was detected. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC5F | ISTDRSFP | An unrecognized combination of node roles was detected. | Full |
| ISTDRC60 | ISTDRSIC | An unrecognized state was received. | Full |
| ISTDRC61 | ISTDRSIC | An unrecognized status was received. | Full |
| ISTDRC62 | ISTDRSIN | An unrecognized search status for a called task was received. | Full |
| ISTDRC63 | ISTDRSMD | An unrecognized function parameter was received. | Full |
| ISTDRC64 | ISTDRSNV | An unrecognized function was specified. | Full |
| ISTDRC65 | ISTDRSNV | An unexpected interprocess signal was received. | Full |
| ISTDRC66 | ISTDRSOH | An unrecognized function code was received. | Full |
| ISTDRC67 | ISTDRSOH | An unrecognized combination of inputs was received. | Full |
| ISTDRC68 | ISTDRSOH | An unrecognized combination of inputs was received. | Full |
| ISTDRC69 | ISTDRSOH | An unexpected interprocess signal was received. | Full |
| ISTDRC6A | ISTDRSOH | An unrecognized state was detected. | Full |
| ISTDRC6B | ISTDRSQS | An unrecognized function was detected. | Full |
| ISTDRC6C | ISTDRSSQ | An unrecognized search task was requested. | Full |
| ISTDRC6D | ISTDRSYB | VTAM was unable to remove a locate control block from the list. | Full |
| ISTDRC6E | ISTDRYCD | An unrecognized query function was detected. | Full |
| ISTDRC6F | ISTDRSCA | An unrecognized entry type was specified. | Full |
| ISTDRC70 | ISTDRDDD | A release of storage associated with a resource failed to complete. | Full |
| ISTDRC71 | ISTDROLO | A build of a Locate CV82 is required, but none is present. | Full |
| ISTDRC72 | ISTDROFI | A build of a Find CV40 is required, but none is present. | Full |
| ISTDRC73 | ISTDRMAC | The input received is not recognized for the current state. | Full |
| ISTDRC74 | ISTDRMAC | The input received is not recognized for the current state. | Full |
| ISTDRC75 | ISTDRMAC | An unrecognized state was received. | Full |
| ISTDRC76 | ISTDRMAC | An unexpected return code was received. | Full |
| ISTDRC77 | ISTDRRCF | An unrecognized state was received. | Full |
| ISTDRC78 | ISTDRRCP | A register GDS variable is required, but is not present. | Full |
| ISTDRC79 | ISTDRRCR | An unexpected return code was received. | Full |
| ISTDRC7A | ISTDRRNE | An unexpected return code was received. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC7B | ISTDRRNF | An unrecognized state was received. | Full |
| ISTDRC7C | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC7D | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC7E | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC7F | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC80 | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC81 | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC82 | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC83 | ISTDRRNF | The input received is not recognized for the current state. | Full |
| ISTDRC84 | ISTDRRNP | A locate control block is required but is not present. | Full |
| ISTDRC85 | ISTDRRNP | An unrecognized state was detected. | Full |
| ISTDRC86 | ISTDRRNR | An unexpected return code was received. | Full |
| ISTDRC87 | ISTDRRNR | The available length is not sufficient for a required GDS variable. | Full |
| ISTDRC88 | ISTDRRNR | The available length is not sufficient for a required GDS variable. | Full |
| ISTDRC89 | ISTDRRNR | An unexpected return code was received. | Full |
| ISTDRC8A | ISTDRRNR | The available length is not sufficient for a required GDS variable. | Full |
| ISTDRC8B | ISTDRSBC | A parsed locate element is required, but is not present. | Full |
| ISTDRC8C | ISTDRSNB | A state that is not valid was detected upon entry to ISTDRSNB. | Full |
| ISTDRC8D | ISTDRSSD | An unknown task was loaded. | Full |
| ISTDRC8E | ISTDRFDS | A directory server notify was received with an empty list. | Full |
| ISTDRC8F | ISTDRFDS | An unexpected return code was received. | Full |
| ISTDRC90 | ISTDRSHR | An unexpected input operation was received. | Full |
| ISTDRC91 | ISTDRSHR | An unrecognized max hierarchy was detected. | Full |
| ISTDRC92 | ISTDRSHR | An unexpected resource CV combination was detected. | Full |
| ISTDRC93 | ISTDRSHR | An unexpected max hierarchy combination was detected. | Full |
| ISTDRC94 | ISTDRSNI | An unrecognized locate was detected. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRC95 | ISTDRSNI | An unrecognized node role was identified in an LCB. | Full |
| ISTDRC96 | ISTDRDUP | A DBUPDATE was attempted at a VTAM end node. | Full |
| ISTDRC97 | ISTDRDQP | A DBQUERY was attempted at a VTAM end node. | Full |
| ISTDRC98 | ISTDRDDP | A DBDELETE was attempted at a VTAM end node. | Full |
| ISTDRC99 | ISTDRERT | The DS PAB router received an unrecognized IPS. | Full |
| ISTDRC9A | ISTDROCG | Unrecognized parameter value was received. | Full |
| ISTDRC9B | ISTDRRNF | Input not valid for current finite state machine (FSM) state | Full |
| ISTDRC9C | ISTDRRNF | Input not valid for current finite state machine (FSM) state | Full |
| ISTDRC9D | ISTDRRNF | Input not valid for current finite state machine (FSM) state | Full |
| ISTDRC9E | ISTDROFI | A build of a FIND CV26 is required, but none is present. | Full |
| ISTDRC9F | ISTDROFO | A build of a FOUND CV26 is required, but none is present. | Full |
| ISTDRCA1 | ISTDREDC | DSME exit returned a central directory server (CDS) list that was not valid. | Full |
| ISTDRCA7 | ISTDRMBO | An unrecognized interprocess signal was received. | Full |
| ISTDRCA8 | ISTDRMBO | An unrecognized interprocess signal was received. | Full |
| ISTDRCA9 | ISTDRMBO | An unrecognized interprocess signal was received. | Full |
| ISTDRCAA | ISTDRMBO | An unrecognized state was detected. | Full |
| ISTDRCA2 | ISTDRMNV | Not valid DLLU_FSM Input. | Full |
| ISTDRCAB | ISTDREBN | An unrecognized state was detected. | Full |
| ISTDRCAC | ISTDRMBN | An unrecognized state was detected. | Full |
| ISTDRCAD | ISTDRMGS | An unrecognized interprocess signal was received. | Full |
| ISTDRCAE | ISTDRMGS | An unrecognized state was detected. | Full |
| ISTDRCAF | ISTDRFGS | An unrecognized control vector was detected; CV0E was expected but is not present. | Full |
| ISTDRCB0 | ISTDRFGS | An unrecognized control vector was detected; CV35 was expected but is not present. | Full |
| ISTDRCB1 | ISTDRFGU | An unrecognized control vector was detected; CV35 was expected but is not present. | Full |

*Table 17. FFST probes for directory services (ISTDRCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTDRCB2 | ISTDROIE | A build of a CV0E failed due to an unrecognized name. | Full |
| ISTDRCB3 | ISTDROIE | A build of a Find CV0E is required, but none is present. | Full |
| ISTDRCB4 | ISTDRFMH | Incorrect hierarchy was passed to module. | Full |
| ISTDRCB5 | ISTDRFBN | A unrecognized network-qualified name was detected. | Full |
| ISTDRCB6 | ISTDRFBN | List operation failed. | Full |
| ISTDRCB7 | ISTDRFBN | A unrecognized network-qualified name was detected. | Full |
| ISTDRCB8 | ISTDRFBN | An attempt to free storage failed. | Full |
| ISTDRCB9 | ISTDRSPC | List operation failed. | Full |
| ISTDRCBA | ISTDRSPC | List operation failed. | Full |
| ISTDRCBB | ISTDRSPC | List operation failed. | Full |
| ISTDRCBC | ISTDRSPC | List operation failed. | Full |
| ISTDRCBD | ISTDRSPC | Attempt to free storage failed. | Full |
| ISTDRCBE | ISTDREIN | An unrecognized state was detected. | Full |
| ISTDRCC1 | ISTDRSDT | An unrecognized state was detected on entry. | Full |
| ISTDRCC2 | ISTDROFI | Build of a find CV4A required, but none were present. | Full |
| ISTDRCC3 | ISTDROFO | Build of a found CV4A required, but none were present. | Full |
| ISTDRCC4 | ISTDRSLT | RDS search request is hung. | Full |
| ISTDRCC5 | ISTDRSOA | Attempted to add an OSCB to a list when it is already on a list. | Full |

## Half-session services probes

*Table 18. FFST probes for half-session services (ISTHSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTHSC01 | ISTTSCUD | A problem occurred trying to free a CSM buffer. | Full |

# Enterprise Extender probes

*Table 19. FFST probes for Enterprise Extender (ISTAUCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTAUC01 | ISTAUCPF | Unusable work element was queued to Enterprise Extender port finite state machine (FSM) processor ISTAUCPF. | Mini |
| ISTAUC02 | ISTAUCPF | Work-in-progress field (IPNCB_WIP) in the Enterprise Extender port control block IPNCB has not been set and should have been set during DACTLINK processing. | Mini |

*Table 19. FFST probes for Enterprise Extender (ISTAUCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTAUC03 | ISTAUCPF | Work element address (WEA) field in the request parameter header control block (RPH) does not point to an AUDP parameter list (AUCPL). | Mini |
| ISTAUC04 | ISTAUCLF | Unusable work element was queued to the Enterprise Extender line finite state machine (FSM) processor ISTAUCLF. | Mini |
| ISTAUC05 | ISTAUCIF | Unusable work element was queued to the Enterprise Extender call-in finite state machine (FSM) processor ISTAUCIF. | Mini |
| ISTAUC06 | ISTAUCOF | Unusable work element was queued to the Enterprise Extender call-out finite state machine (FSM) processor ISTAUCOF. | Mini |
| ISTAUC07 | ISTAUCCF | Unusable work element was queued to the Enterprise Extender connection processor ISTAUCCF. | Mini |
| ISTAUC08 | ISTAUCOL | An XID signal that is not valid was received by the outbound signal processor ISTAUCOL. | Mini |
| ISTAUC09 | ISTAUCIL | An XID signal that is not valid was received by the inbound signal processor ISTAUCIL. | Mini |

# Logical link control

*Table 20. FFST probes for logical link control*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTIUT02 | IUTLLCII | Primitive is not valid. | Mini |
| ISTIUT03 | IUTLLCI0 | Primitive is not valid. | Mini |
| ISTIUT04 | IUTLLCRD | Unrecognized primitive. | Mini |
| ISTIUT05 | IUTLLCRD | Input is not a response. | Mini |
| ISTIUT06 | IUTALCII | Primitive is not valid. | Mini |
| ISTIUT14 | IUTLLC9D | Unrecognized primitive. | Mini |
| ISTIUT15 | IUTLLC9D | Input is not a valid request. | Mini |
| ISTIUT38 | CMLOC$AC | Act_SAP_CNF - state error | Mini |
| ISTIUT42 | CMLOC$EC | Enable_IC_Cnf - state error | Mini |
| ISTIUT44 | CMLOC$EI | Enable_IC_Ind - state error | Mini |
| ISTIUT49 | CMLOC$SC | Call_Setup_Cnf - state error | Mini |
| ISTIUT50 | CMLOC$TC | Deactivate_SAP_Cnf - state error | Mini |
| ISTIUT51 | CMLOC$SI | Call_Setup_Ind - state error | Mini |
| ISTIUT65 | REGMGSRK | Resolve_Token - object address is zero | Mini |
| ISTIUT66 | REGMGRUK | Reuse_Token - token is not valid | Mini |
| ISTIUT67 | REGMGRIK | Invalidate_Token - object address is zero | Mini |
| ISTIUT72 | IUTLLCRM | Registration manager initialization FSM error | Mini |

*Table 20. FFST probes for logical link control  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTIUT73 | ISTLLCM8 | Unusable input has been received by IDX GROUP CONTROL (LLCM8). | Mini |
| ISTIUT74 | ISTLLCM8 | IDX GROUP CONTROL (LLCM8) was unable to obtain an RU processing element (RUPE) for a critical identification exchange (IDX) function. | Mini |
| ISTIUT75 | ISTLLCM8 | IDX GROUP CONTROL (LLCM8) received an unexpected RU processing element. | Mini |
| ISTIUT76 | IUTLLCTP | Lost PDUS - The READ side of the HPDT connection detected a lost packet. | Full |

## LU network services probes

*Table 21. FFST probes for LU network services (ISTNSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTNSC01 | ISTNSCRS | A suspended log mode is not found.<br><br>Check the mode the application program specifies on the APPCCMD CONTROL=OPRCNTRL, QUALIFY=RESTORE macroinstruction.<br><br>You can use the following VTAM dump functions to display advanced program-to-program communication (APPC)-related information:<br>• APPLCONV<br>• APPLMODE<br>• APPMODAL<br>• PARTNRLU | Full |

## LU resource manager probes

*Table 22. FFST probes for LU resource manager (ISTRMCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTRMC01 | ISTRMCLR | A QUERY_SNASVCMG response was received, but none was outstanding. | Full |
| ISTRMC02 | ISTRMCLR | The mode name for the specified LU was not found in the LU 6.2 logmode table. | Full |
| ISTRMC03 | ISTRMCTP | The mode name for the specified LU was not found in the LU 6.2 logmode table. | Full |

## Management services transport probes

*Table 23. FFST probes for management services transport (ISTMTCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTMTC01 | ISTMTCFF | The main router received an unrecognized request-unit processing element. | Mini |

*Table 23. FFST probes for management services transport (ISTMTCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTMTC02 | ISTMTCFF | The main router received an unrecognized work element. | Full |
| ISTMTC03 | ISTMTCFF | The START_TP_REPLY signal that is received contains a return code that is incorrect or unrecognized for this leg. | Mini |
| ISTMTC04 | ISTMTCFF | An unexpected condition is detected when a query request is sent to the NetView application program's programmable peripheral interface (PPI) queue. | Mini |
| ISTMTC05 | ISTMTCFF | An unexpected condition is detected while defining the VTAM queue to the NetView application program's programmable peripheral interface (PPI). | Mini |
| ISTMTC06 | ISTMTCFF | An unexpected condition is detected while sending data from VTAM to the NetView processor across the programmable peripheral interface (PPI). | Mini |
| ISTMTC07 | ISTMTCFF | The main router received an unrecognized work element. | Mini |
| ISTMTC08 | ISTMTCFF | Unrecognized data is received from the NetView application program across the programmable peripheral interface (PPI). | Mini |
| ISTMTC09 | ISTMTCFF | Unrecognized data is received from the NetView application program. | Mini |
| ISTMTC10 | ISTMTCFF | The PPI queue received from the NetView application program contains unrecognized data. | Mini |
| ISTMTC11 | ISTMTCFF | VTAM could not create a process scheduling table for MI. | Mini |
| ISTMTC12 | ISTMTCFF | The MDS-MU received from the NetView application program is unrecognized. | Mini |
| ISTMTC13 | ISTMTCFF | An expected outstanding request unit is not found on the outstanding request unit list. | Mini |
| ISTMTC14 | ISTMTCFF | An expected outstanding request unit is not found on the outstanding request unit list. | Mini |
| ISTMTC15 | ISTMTCFF | An expected session list entry is not found, but an ATL entry is found. | Mini |
| ISTMTC16 | ISTMTCFF | An expected session-list entry is not found, but an ATL entry is found. | Mini |

## Network resource management probes

*Table 24. FFST probes for network resource management (ISTNACxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTNAC01 | ISTNACTT | VTAM cannot complete a normal CLOSE ACB and must force the CLOSE ACB. | Full |

## Presentation services probes

*Table 25. FFST probes for presentation services (ISTPSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTPSC01 | ISTPSCMD | CSM pool control block corrupted. | Mini |
| ISTPSC02 | ISTPSCMD | A pool token was specified that is not valid. | Mini |
| ISTPSC03 | ISTPSCMD | CSM detected an MVS system error. | Mini |
| ISTPSC04 | ISTPSCMD | An unexpected CSM reason code was encountered. | Mini |
| ISTPSC06 | ISTPSCFR | A buffer token was specified that is not valid. | Mini |
| ISTPSC07 | ISTPSCFR | CSM detected an MVS system error. | Mini |
| ISTPSC08 | ISTPSCFR | An unexpected CSM reason code was encountered. | Mini |
| ISTPSC09 | ISTPSCFP | A buffer token was specified that is not valid. | Mini |
| ISTPSC10 | ISTPSCFP | CSM detected an MVS system error. | Mini |
| ISTPSC11 | ISTPSCFP | An unexpected CSM reason code was encountered. | Mini |
| ISTPSC12 | ISTPSCMD | Data resides in a fixed I/O buffer instead of a CSM buffer. | Full |
| ISTPSC13 | ISTPSCBM | Unexpected CSM reason code. | Full |
| ISTPSC14 | ISTPSCMF | Unexpected CSM reason code. | Full |

## Recovery manager (RV) probes

The field RVM_FFST_ID within the ISTRVM control block contains additional information about the location from which the FFST PROBE was triggered.

*Table 26. FFST probes for recovery manager (ISTRVMxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTRVM00 | ISTRVMRT | Unexpected condition | Mini |
| ISTRVM01 | ISTRVMRT | Unexpected RUPE | Mini |
| ISTRVM02 | ISTRVMRT | Unexpected Data Recovered signal | Mini |
| ISTRVM03 | ISTRVMIN | Unexpected condition | Mini |
| ISTRVM04 | ISTRVMOS | Unexpected Data Recovered signal | Mini |
| ISTRVM05 | ISTRVMCR | Unexpected condition | Mini |
| ISTRVM06 | ISTRVMDL | Unexpected condition | Mini |
| ISTRVM07 | ISTRVMDL | Unexpected condition | Mini |
| ISTRVM08 | ISTRVMDS | Unexpected condition | Mini |
| ISTRVM0A | ISTRVRPI | Unknown RUPE from RCM | Mini |
| ISTRVM0B | ISTRVRPI | Incorrect state | Mini |
| ISTRVM0C | ISTRVRPI | Received RTPALS in wrong FSM state | Mini |
| ISTRVM0D | ISTRVRPI | Received COMPLETE in wrong FSM state | Mini |
| ISTRVM0E | ISTRVRPQ | Unknown RUPE from SS | Mini |
| ISTRVM0F | ISTRVRPQ | Unexpected CV on sessinfo rsp | Mini |

*Table 26. FFST probes for recovery manager (ISTRVMxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTRVM10 | ISTRVRCM | Unexpected failure from TSCM5 | Mini |
| ISTRVM11 | ISTTSCM2 | Unexpected control block in chain | Mini |
| ISTRVM12 | ISTTSCM3 | Unexpected control block in chain | Mini |
| ISTRVM13 | ISTRVRCM | Unexpected failure from GETENTRY | Mini |
| ISTRVM14 | ISTRVMXF | Unexpected failure | Full |
| ISTRVM15 | ISTTSM1D | TREE ADD failed | Full |
| ISTRVM16 | ISTTSM1E | TREE ADD failed | Full |
| ISTRVM17 | ISTRVMLM | LM table restore failure | Full |
| ISTRVM18 | ISTRVRPI | Received APPCINFO in wrong FSM | Full |
| ISTRVM19 | ISTTSM1A | Duplicate FID5 address in HIT | Full |

## Session services probes

*Table 27. FFST probes for session services (ISTSSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSSC01 | ISTSSCTM | RU parsing error. | Mini |
| ISTSSC02 | ISTSSCXM | Address conflict was detected during merge. | Mini |
| ISTSSC03 | ISTSSCXV | An SRT was missing for an autologon session. As a result, session setup failed. | Full |
| ISTSSC04 | ISTSSCFC | FREEBLK failure | Full |
| ISTSSC05 | ISTSSCXY | FREEBLK failure | Full |
| ISTSSC06 | ISTSSCKI | FREEBLK failure | Full |
| ISTSSC07 | ISTSXCQ1 | No session control block could be located for the FQPCID supplied by the recovery PAB on the multinode persistent Session Started signal flow. | Full |
| ISTSSC08 | ISTSXCU5 | Session services could not locate the correct recovery PAB when attempting to send a response to the recovery PAB's request to recover a multinode persistent session. | Full |
| ISTSSCZ1 | ISTSSCYC | An attempt was made to dequeue an SIB from a PU's SIB queue, but the SIB was not on the queue. | Full |
| ISTSSCZ2 | ISTSSCZY | A loop was detected in ISTSSCZY. | Full |

## Session services CP-CP probes

*Table 28. FFST probes for session services CP-CP (ISTSCCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSCC01 | ISTSCCRT | A request processing element is received that contained an unrecognized request unit processing element (RUPE). | Mini |

*Table 28. FFST probes for session services CP-CP (ISTSCCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSCC02 | ISTSCRCA | SSC attempted to add an adjacent node control block to the adjacent node control block list, but the add failed to complete normally. | Mini |
| ISTSCC03 | ISTSCCFM | The SSC coordinator finite state machine processor received null input in an unexpected state when one or more of the CP-CP sessions was in active or pending active status. | Mini |
| ISTSCC04 | ISTSCUAR | An internal SSC module passed an unrecognized session status to ISTSCUAR. | Mini |
| ISTSCC05 | ISTSCUAR | An internal SSC module passed an unrecognized session type to ISTSCUAR. | Mini |
| ISTSCC06 | ISTSCUBS | An internal SSC module passed an unrecognized session type to ISTSCUBS. | Mini |
| ISTSCC07 | ISTSCUBS | An internal SSC module passed an unrecognized session status to ISTSCUBS. | Mini |
| ISTSCC08 | ISTSCUDS | An internal SSC module passed an unrecognized session type to ISTSCUDS. | Mini |
| ISTSCC09 | ISTSCUNO | SSC entered an unexpected state on a contention-winner session. | Mini |
| ISTSCC10 | ISTSCUNO | SSC entered an unexpected state on a contention-loser session. | Mini |
| ISTSCC11 | ISTSCUNO | An internal SSC module passed an unrecognized session type to ISTSCUNO. | Mini |
| ISTSCC15 | ISTSCCFM | The SSC coordinator finite state machine processor received nonnull input in an unexpected state when one or more of the CP-CP sessions is in active or pending active status. | Mini |
| ISTSCC16 | ISTSCTRQ | The REQUEST_CP_CAPABILITIES_TP received as input a signal that is neither an ACT_CP_CP_SESSION request nor a CONTINUE_CW response. | Mini |
| ISTSCC19 | ISTSCTRC | The SEND_REJECT_CONVGRP_TP received an input signal other than a DEACTIVATE_CP_CP_SESSION request. | Mini |
| ISTSCC20 | ISTSCCFM | The SSC coordinator finite state machine processor encountered a should-not-occur condition and no CP-CP session cleanup is required. | Mini |
| ISTSCC21 | ISTSCCRT | An adjacent node control block is not found for the received signal. | Full |
| ISTSCC23 | ISTSCCRT | A processing element is queued to session services CP-CP (SSC) but is not recognized to be either a request unit processing element (RUPE) or a timer queue element. | Mini |

## Session services LU-LU probes

The following table contains the LU-LU session services (SSL) software probes. When an SSL software probe is triggered, an error has occurred while processing network session services for an LU-LU connection. The search, session initiation, or request causing the error is rejected. VTAM will attempt to recover from the error and continue processing subsequent requests. Recovery might not be possible for some types of errors and subsequent requests might fail, terminals might hang, and other types of errors might occur.

A system dump is taken to assist in identifying the source of the problem. A VABEND macroinstruction can also be processed if specified by the probe instruction. The processing element is freed and processing continues.

*Table 29. FFST probes for session services LU-LU (ISTSLCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC01 | ISTSLCRT | An unknown interprocess signal was dispatched to the SSL component process anchor block (PAB). | Mini |
| ISTSLC02 | ISTSLRDP | Finite state machine information is missing when a SEARCH_RPY signal is processed. | Mini |
| ISTSLC03 | ISTSLRDP | Finite state machine information is missing when a CDINIT SEARCH_RPY signal is processed. | Mini |
| ISTSLC04 | ISTSLRDP | Finite state machine information is missing when an IOCD SEARCH_RPY signal is processed. | Mini |
| ISTSLC05 | ISTSLRDP | Finite state machine information is missing when a NOTIFY SEARCH_RPY signal is processed. | Mini |
| ISTSLC06 | ISTSLRDP | Finite state machine information is missing when a SEARCH_RPY signal is processed and a storage failure occurs. | Mini |
| ISTSLC07 | ISTSLRXS | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC08 | ISTSLTPI | An unexpected combination of conditions was encountered during third-party initiated finite state machine processing. | Mini |
| ISTSLC09 | ISTSLBXN | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC10 | ISTSLURR | Route Selection control vector (RSCV) information is missing from SSL's session initiation control block when performing a build request for a RECOMPUTE_ROUTE signal. | Mini |
| ISTSLC11 | ISTSLBXJ | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC12 | ISTSLRXJ | Finite state machine information is missing when an INIT_OTHER_COMP signal is processed, and a storage failure occurred. | Mini |

*Table 29. FFST probes for session services LU-LU (ISTSLCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC13 | ISTSLRXJ | Finite state machine information is missing when an INIT_OTHER_COMP signal is processed. | Mini |
| ISTSLC14 | ISTSLBXI | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC15 | ISTSLUDS | An error occurred while the SSL component was removing a session initiation control block key from the list. | Full |
| ISTSLC16 | ISTSLUDS | An error occurred while the SSL component was removing a session initiation control block from the list. | Mini |
| ISTSLC17 | ISTSLUGT | The ACMDT's node type information for this node had an unknown value because it did not indicate whether this is a VTAM end node or a VTAM network node. | Mini |
| ISTSLC18 | ISTSLUBX | A build request was received for an unrecognized interprocess signal. | Mini |
| ISTSLC19 | ISTSLUBX | A build request was received for an unrecognized interprocess signal. | Mini |
| ISTSLC20 | ISTSLUBX | The SSL component could not remove a fully qualified procedure correlation identifier (FQPCID) from the list. | Mini |
| ISTSLC21 | ISTSLUBX | The SSL component could not remove a FQPCID_MODIFIER from the list. | Mini |
| ISTSLC22 | ISTSLRXI | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC23 | ISTSLVPE | An unrecognized signal was received from directory services. | Mini |
| ISTSLC24 | ISTSLUSC | The SSL component was unable to determine the type of signal to send to session services. | Mini |
| ISTSLC25 | ISTSLURX | An unrecognized interprocess signal was received from session services. | Mini |
| ISTSLC26 | ISTSLURD | An unrecognized interprocess signal was received from directory services. | Mini |
| ISTSLC27 | ISTSLSNO | The Route Selection control vector (RSCV) information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC28 | ISTSLSNO | The Route Selection control vector (RSCV) information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC29 | ISTSLSNO | An unrecognized session initiation status was received for the current system state. | Mini |

## FFST probes

*Table 29. FFST probes for session services LU-LU (ISTSLCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC30 | ISTSLSNO | An unexpected combination of conditions was encountered during finite state machine processing initiated by the secondary logical unit (SLU) at a network node (NN). | Full |
| ISTSLC31 | ISTSLUBD | A build request was received for an unrecognized interprocess signal. | Mini |
| ISTSLC32 | ISTSLSED | An unexpected combination of conditions was encountered during finite state machine processing initiated by the secondary logical unit (SLU) at an end node (EN) destination logical unit (DLU). | Mini |
| ISTSLC33 | ISTSLRXO | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC34 | ISTSLRXQ | Finite state machine information is missing when a QUEUED signal is processed. | Mini |
| ISTSLC35 | ISTSLRXC | Finite state machine information is missing when a SESS_REQ_COMP signal is processed. | Mini |
| ISTSLC36 | ISTSLRXC | Finite state machine information is missing when a SESS_REQ_COMP signal is processed and storage problems occur. | Mini |
| ISTSLC37 | ISTSLBXV | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC38 | ISTSLBXS | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC39 | ISTSLRXD | Finite state machine information is missing when a DEQUEUE signal is processed. | Mini |
| ISTSLC40 | ISTSLPND | An unexpected combination of conditions was encountered during finite state machine processing initiated by the primary logical unit (PLU) at a network node. | Mini |
| ISTSLC41 | ISTSLRXV | Finite state machine information is missing when a PROV_SC signal is processed. | Mini |
| ISTSLC42 | ISTSLPNO | The Route Selection control vector (RSCV) information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC43 | ISTSLPNO | The RSCV information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC44 | ISTSLPNO | An unexpected combination of conditions was encountered during finite state machine processing initiated by the primary logical unit (PLU) at a network node. | Full |
| ISTSLC45 | ISTSLPNO | The RSCV information is missing from the control vector list of the session initiation control block. | Mini |

*Table 29. FFST probes for session services LU-LU (ISTSLCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC46 | ISTSLBXQ | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC47 | ISTSLBXC | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC48 | ISTSLRTT | Finite state machine information is missing when a RECOMPUTE_ROUTE_RPY signal is processed. | Mini |
| ISTSLC49 | ISTSLRTR | Finite state machine information is missing when a REQ_ROUTE_RPY signal is processed. | Mini |
| ISTSLC50 | ISTSLRDR | Finite state machine information is missing when a SEARCH_REQ signal is processed. | Mini |
| ISTSLC51 | ISTSLRTC | Class of service (COS) and transmission priority field (TPF) control vector information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC52 | ISTSLRTC | Finite state machine information is missing when a REQ_COS_TPF_RPY signal is processed. | Mini |
| ISTSLC53 | ISTSLRTV | Finite state machine information is missing when a REQ_TG_VECTORS_RPY signal is processed. | Mini |
| ISTSLC54 | ISTSLRNT | The NOTIFY generalized data stream (GDS) variable being processed contained a NOTIFY_TYPE that was unknown or not valid. | Mini |
| ISTSLC55 | ISTSLRNT | Notify control vector 80 (CV80) information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC56 | ISTSLRNT | Notify CV81 information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC57 | ISTSLRDN | The CDINIT GDS variable information is missing from a NON_VERIFY_REQ signal received from directory services. | Mini |
| ISTSLC58 | ISTSLRDC | Finite state machine information is missing when a CHAIN_FLOW signal is processed. | Mini |
| ISTSLC59 | ISTSLPXR | An unexpected combination of conditions was encountered during processing of the PLU-initiate exit border node preprocessor finite state machine (FSM). | Mini |
| ISTSLC60 | ISTSLPXT | An unexpected combination of conditions was encountered during processing of the PLU-initiate exit border node postprocessor finite state machine. | Mini |

*Table 29. FFST probes for session services LU-LU (ISTSLCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC61 | ISTSLPYR | An unexpected combination of conditions was encountered during processing of the PLU-initiate entry border node preprocessor finite state machine. | Mini |
| ISTSLC62 | ISTSLPNS | RSCV information is missing from the control vector list of the session initiation control block. | Mini |
| ISTSLC63 | ISTSLPNS | An unexpected combination of conditions was encountered during finite state machine processing initiated by the primary logical unit (PLU) at a network node. | Mini |
| ISTSLC64 | ISTSLPED | An unexpected combination of conditions was encountered during finite state machine processing initiated by the primary logical unit (PLU) at an end node destination logical unit. | Mini |
| ISTSLC65 | ISTSLPEO | An unexpected combination of conditions was encountered during finite state machine processing initiated by the primary logical unit (PLU) at an end node origination logical unit. | Mini |
| ISTSLC66 | ISTSLBXT | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC67 | ISTSLBXR | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC68 | ISTSLBXP | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC69 | ISTSLBXO | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC70 | ISTSLBXE | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC71 | ISTSLBXD | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC72 | ISTSLBDL | A failure occurred while performing postprocessor information processing for a signal being sent to session services. | Mini |
| ISTSLC73 | ISTSLSND | An unrecognized session initiation status was received for the current system state in SLU-initiate finite state machine processing. | Mini |
| ISTSLC74 | ISTSLSND | An unexpected combination of conditions occurred during SLU-initiate finite state machine processing at a network node. | Mini |

*Table 29. FFST probes for session services LU-LU (ISTSLCxx)  (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC75 | ISTSLSEO | An unexpected combination of conditions occurred during SLU-initiate finite state machine processing at an end node origination logical unit. | Mini |
| ISTSLC76 | ISTSLUCS | The SSL component was unable to remove the key entry for the session initiation control block from the list. | Full |
| ISTSLC77 | ISTSLRXN | Finite state machine information is missing when a PEND_SC signal is processed. | Mini |
| ISTSLC78 | ISTSLUSL | An unrecognized LU role was provided for a session initiation control block find request. | Mini |
| ISTSLC79 | ISTSLCRT | An unrecognized work element was dispatched to the SSL process anchor block (PAB). | Mini |
| ISTSLC80 | ISTSLR63 | The SSL component was unable to remove a CV63 from the control vector list of the session initiation control block. | Mini |
| ISTSLC81 | ISTSLBLC | A CV82 control vector contains a network-qualified sender name that is not valid. | Mini |
| ISTSLC82 | ISTSLBLC | A CV82 control vector contains a network-qualified host receiver name that is not valid. | Mini |
| ISTSLC84 | ISTSLUNV | The original class of service (COS) associated with the SLU-initiate request was not found in the parent session initiation control block during a nonverify search sent in response to a SLU-initiate request. | Mini |
| ISTSLC85 | ISTSLRDI | The SSL component received a SESS_INIT_INFO_REQ signal which contained inconsistent data. | Full |
| ISTSLC86 | ISTSLBXB | Failure occurred when performing front-end processing for a BN_SESS_RPY signal being sent to session services. | Mini |
| ISTSLC87 | ISTSLBXH | Failure occurred when performing front-end processing for a CACHE_BN_INFO signal being sent to session services. | Mini |
| ISTSLC88 | ISTSLBTM | The network-qualified adjacent nonnative CP name is not valid. | Mini |
| ISTSLC89 | ISTSLRXB | The network-qualified adjacent LU name is not valid. | Mini |
| ISTSLC90 | ISTSLRXB | Failure occurred when performing front-end processing for a BN_SESS_RPY signal being sent to session services. | Mini |
| ISTSLC91 | ISTSLRTM | Finite state machine information was missing when a PROCESS_MAP_COS signal from topology routing services (TRS) was being processed. | Mini |

*Table 29. FFST probes for session services LU-LU (ISTSLCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLC92 | ISTSLUBX | The RSCV that should have been in the session initiation control block when building a CACHE_BN_INFO to session services was missing. | Mini |
| ISTSLC93 | ISTSLRXB | The class of service (COS) was received on a BN_SESS_REQ, but the primary logical unit (PLU) COS from LOCATE time processing could not be found. | Mini |
| ISTSLC94 | ISTSLPYT | An unexpected combination of conditions was encountered during processing of the PLU-initiate entry border node postprocessor finite state machine. | Mini |
| ISTSLC95 | ISTSLSXR | An unexpected combination of conditions was encountered during processing of the SLU-initiate exit border node preprocessor finite state machine. | Mini |
| ISTSLC96 | ISTSLSXT | An unexpected combination of conditions was encountered during processing of the SLU-initiate exit border node postprocessor finite state machine. | Mini |
| ISTSLC97 | ISTSLSYR | An unexpected combination of conditions was encountered during processing of the SLU-initiate entry border node preprocessor finite state machine. | Mini |
| ISTSLC98 | ISTSLSYT | An unexpected combination of conditions was encountered during processing of the SLU-initiate entry border node postprocessor finite state machine. | Mini |
| ISTSLC99 | ISTSLUBR | An unexpected combination of conditions was encountered during border node request finite state machine processing. | Mini |
| ISTSLCA0 | ISTSLUFR | The finite state machine router was called for a situation where a finite state machine (FSM) is not used. | Mini |
| ISTSLCA1 | ISTSLSNO | Route Selection Control Vector (RSCV) missing from the session initiation control block's control vector list. | Mini |
| ISTSLCA2 | ISTSLUAS | An unexpected combination of conditions was encountered during dependent LU requester search finite state machine processing. | Full |
| ISTSLCA3 | ISTSLR2B | Missing subvector on CV2B (RSCV). | Full |
| ISTSLCA4 | ISTSLRDI | The DLU is nonnative, but the previously stored DLU information and the DLU information on the SESS_INIT_INFO_REQ IPS just received do not match. | Mini |
| ISTSLCA5 | ISTSLRDP | The DLU is nonnative, but the previously stored DLU information and the DLU information on the SEARCH_RPY IPS just received do not match. | Mini |

*Table 29. FFST probes for session services LU-LU (ISTSLCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTSLCA7 | ISTSLRDSD | A call to ISTSXCXS failed, either unknown RU or insufficient storage. | Mini |
| ISTSLCA8 | ISTSLU85 | SSLCNO action invoked in FSMe. | Mini |

## Storage management probes

*Table 30. FFST probes for storage management (ISTORCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTORC01 | ISTORCHB | Unexpected I/O condition was detected. The output contains the buffer pool control block (BPCB) and one page of storage from each buffer pool expansion block (PXB). | Full |
| ISTORCZ1 | ISTORCGB | Issued when an overlay of a GETBLKed header is detected and before the element is dequeued from the FBQE. | Full |
| ISTORCZ2 | ISTORCFB | GETBLD header overlay | Full |

## Topology and routing services probes

*Table 31. FFST probes for topology and routing services (ISTTRCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTRC01 | ISTTRCRT | An unknown work element is received. | Mini |
| ISTTRC02 | ISTTRCRT | An unknown signal is contained with the work element. | Mini |
| ISTTRC03 | ISTTRQDN | Issued when attempting to clean up storage for a node information block that is not found. | Full |
| ISTTRC04 | ISTTRIFP | While processing a topology database update (TDU), an unrecognized finite state machine state is encountered. | Mini |
| ISTTRC05 | ISTTRINP | An internal parsing error has occurred. An Input_TDU is encountered with a mismatch between the number of node vectors and the number of associated transmission group (TG) vectors. | Mini |
| ISTTRC06 | ISTTRPSH | An unrecognized transmission group (TG) type is received on a RQ_SINGLE_HOP_ROUTE signal. | Mini |
| ISTTRC07 | ISTTRQDT | Unable to delete a node from a tree because the node cannot be found on the tree. | Full |
| ISTTRC08 | ISTTRTLT | An unrecognized partner node type is received on a TG_UPDATE signal. | Mini |
| ISTTRC09 | ISTTRTLT | The transmission group (TG) direction specified on a TG_UPDATE signal is not recognized. | Mini |

*Table 31. FFST probes for topology and routing services (ISTTRCxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTRC10 | ISTTRRPP | A problem is encountered while attempting to uncache the session trees from the topology and routing services (TRS) database. | Full |
| ISTTRC11 | ISTTRRPP | While scanning the topology and routing services (TRS) database an unidentified resource is found. | Mini |
| ISTTRC12 | ISTTRDGM | An unrecognized secondary LU (SLU) node identification is received on a REQ_MULTIPLE_ROUTES signal. | Mini |
| ISTTRC13 | ISTTRRRR | A route to a gateway node is received on a REQUEST_ROUTE; however, the signal contained unrecognized information. | Mini |
| ISTTRC14 | ISTTRATE | Unidentified resource is in database. | Mini |
| ISTTRCZ1 | ISTTRRGT | Class of service information or node information not found. | Full |
| ISTTRCZ2 | ISTTRCLT | Element is not on list. | Full |

## Trace services probes

*Table 32. FFST probes for trace services (ISTRACxx)*

| Probe name | Module | Description | Dump yype |
|---|---|---|---|
| ISTRAC01 | ISTRACZT | FFST dump from the VTAM trap module. | Full |
| ISTRAC02 | ISTRACZT | FFST dump from the VTAM trap module consisting of the ATCVT and the VTAM internal trace (VIT). | Mini |
| ISTRACZ3 | ISTITCAS | FFST full dump for a predetermined APPC sense code. | Full |
| ISTRACZ4 | ISTITCAS | FFST full dump for a predetermined RPL6 return code. | Full |

## Transmission control — Data link control (TC-DLC) probes

*Table 33. FFST probes for TC-DLC (ISTTCCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTCC01 | ISTTCCTD | Multiple function keywords were encountered. TCP-DLC supports ACTPATH, DACTPATH, OPENPATH, and CLOSEPATH. | Full |
| ISTTCC02 | ISTTCCRI | Registration Manager or the DLC sent an unrecognizable RUPE. | Mini |
| ISTTCC03 | IUTLLCID | Unrecognized primitive. | Mini |
| ISTTCC04 | ISTTCCRI | An unexpected RUPE was encountered when a RUPE was sent to Registration Manager. | Full |
| ISTTCC05 | IUTLLCID | Input is not a RUPE response. | Mini |

## Transmission subsystem probes

*Table 34. FFST probes for transmission subsystem (ISTTSCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTSC01 | ISTTSCRI | Inbound path information unit (PIU) with an out-of-order sequence number was detected. The output includes the transmission subsystem control block (TSCB) containing the PIU with the unexpected sequence number and the virtual route control block (VRBLK) containing the expected sequence number. | Mini |
| ISTTSC02 | ISTTSC8S | Unrecognized multipath channel sweep work element was detected. | Mini |
| ISTTSC03 | ISTTSC8E | Unexpected multipath channel sweep finite state machine state was detected. | Mini |
| ISTTSC04 | ISTTSC8I | Zero active read subchannel count was detected. | Mini |
| ISTTSC05 | ISTTSC8E | Zero active write subchannel count was detected. | Mini |
| ISTTSC06 | ISTTSC8E | Zero read sweep count decremented. | Mini |
| ISTTSC07 | ISTTSC8E | Zero write sweep count decremented. | Mini |
| ISTTSC08 | ISTTSC9L | Unexpected input in current state was received. | Mini |
| ISTTSC09 | ISTTSC9C | Unexpected input in current state was received. | Mini |
| ISTTSC10 | ISTTSC8E | Multipath channel incorrect path information unit (PIU) was built. | Mini |
| ISTTSC11 | ISTTSC8D | Unexpected input was received. | Mini |
| ISTTSC12 | ISTTSC8E | Multipath channel group lost the last read device. | Mini |
| ISTTSC13 | ISTTSC8E | An attempt to send a data element response failed. | Mini |
| ISTTSC14 | ISTTSC8E | Multipath channel device busy was detected. | Mini |
| ISTTSC15 | ISTTSCM8 | Multipath channel PU services process anchor block (PUPAB) finite state machine error was detected. | Mini |
| ISTTSC16 | ISTTSCM8 | Unable to obtain storage for a request unit processing element (RUPE). | Mini |
| ISTTSC17 | ISTTSC8X | Multipath channel PUPAB finite state machine error was detected. | Mini |
| ISTTSC18 | ISTTSC8X | Unable to obtain storage for a request unit processing element (RUPE). | Mini |
| ISTTSC19 | ISTTSC8B | Empty node control block (NCB) pending queue was detected. | Mini |
| ISTTSC20 | ISTTSC87 | Unexpected value for request/response unit (RU) was received. | Mini |
| ISTTSC21 | ISTTSC9X | Unexpected input in current state was received. | Mini |

**FFST probes**

Table 34. FFST probes for transmission subsystem (ISTTSCxx)  (continued)

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTSC22 | ISTTSC9P | Unexpected input was received. | Mini |
| ISTTSC23 | ISTTSC9D | Unexpected input was received. | Mini |
| ISTTSC24 | ISTTSC8I | Unexpected multipath channel sweep finite state machine state was detected. | Mini |
| ISTTSC25 | ISTTSC8I | Zero active write subchannel count was detected. | Mini |
| ISTTSC26 | ISTTSC8I | Zero read sweep count decremented. | Mini |
| ISTTSC27 | ISTTSC8I | Zero write sweep count decremented. | Mini |
| ISTTSC28 | ISTTSCM8 | Unexpected input was received. | Mini |
| ISTTSC29 | ISTTSC8X | Unexpected input was received. | Mini |
| ISTTSC30 | ISTTSC6F | ISTTSC6F fails to dequeue the TRGCB from ATCNCBQ during deactivation of a major node. This can result in an abend in ISTTSCWU. | Mini |
| ISTTSC31 | ISTTSCUD | A problem occurred trying to free a CSM buffer. | Full |
| ISTTSC32 | ISTTSC8E | Zero active write subchannel count was detected. | Mini |
| ISTTSC33 | ISTTSC8E | Multipath channel incorrect path information unit (PIU) was built. | Mini |
| ISTTSC34 | ISTTSC8E | An attempt to send a data element response failed. | Mini |
| ISTTSC35 | ISTTSC8E | Multipath channel device busy was detected. | Mini |
| ISTTSC36 | ISTTSCYD | Sense code 08150004 received. | Full |
| ISTTSC37 | ISTTSCYD | Sense code 08090033 received. | Full |
| ISTTSC38 | ISTBSCUB | Sense code 08150004 received. | Full |
| ISTTSC39 | ISTBSCUB | Sense code 08090033 received. | Full |
| ISTTSCZ1 | ISTTSCON | Negative NCBUSECT. | Mini |

## VTAM topology agent probes

Table 35. FFST probes for the VTAM topology agent (ISTTACxx)

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTAC01 | ISTTACOS | The topology agent received a CMIP message string with a syntax error. The topology agent cannot process or respond to the message. | Mini |
| ISTTAC03 | ISTTAIGP | The topology agent received a CMIP GET request with a syntax error. The topology agent cannot process or respond to the request. | Mini |
| ISTTAC04 | ISTTAIDN | The topology agent received a distinguished name with a syntax error in a CMIP request. The topology agent cannot process or respond to the request. | Mini |

*Table 35. FFST probes for the VTAM topology agent (ISTTACxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTAC05 | ISTTACII | The topology agent received a request for initialization at a time when the topology agent was in an internal state other than inactive state. | Mini |
| ISTTAC06 | ISTTACII | The topology agent received a register-accept acknowledgment at a time when the topology agent was in an internal state other than initializing. | Mini |
| ISTTAC07 | ISTTACRT | The topology agent was dispatched with a work element type that the topology agent does not support, such as a work element other than RUPE or TQE. | Mini |
| ISTTAC08 | ISTTACRT | The topology agent was dispatched with a RUPE work element containing an OPCODE that is not supported by the topology agent. | Mini |
| ISTTAC09 | ISTTACRT | The topology agent received a message from CMIP services with an unknown API message type in the API header section of the message. | Mini |
| ISTTAC10 | ISTTACOS | The topology agent received a CMIP message that was not identified as type ROIVapdu. The CMIP message was not a request or linked reply. It was identified as one of the following, which the topology agent does not support:<br>• RORSapdu<br>• ROERapdu<br>• RORJapdu | Mini |
| ISTTAC11 | ISTTAIPA | The topology agent received a CMIP message with a syntax error in one of the following:<br>• Object class<br>• Object instance<br>• Access control<br>• Synchronization<br>• Scope<br>• Filter | Mini |
| ISTTAC12 | ISTTAOSP | The topology agent received a message from configuration services or topology and routing services indicating that a snapshot was in initial data phase when that snapshot was in a phase other than initial data phase. | Mini |
| ISTTAC13 | ISTTAINP | The topology agent received a CMIP CANCEL-GET message with a syntax error. | Mini |
| ISTTAC14 | ISTTAOLT | The topology agent has written a SNAlocaltopology response string in an output buffer. The string is greater than the size of the output buffer and has overlayed other storage. | Mini |

## FFST probes

*Table 35. FFST probes for the VTAM topology agent (ISTTACxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTAC16 | ISTTAOAN | The topology agent has written a SNAnetwork response string in an output buffer. The string is greater than the size of the output buffer and has overlayed other storage. | Mini |
| ISTTAC17 | ISTTAOSN | The topology agent has written a SNAnetwork response string in an output buffer. The string is greater than the size of the output buffer and has overlayed other storage. | Mini |
| ISTTAC18 | ISTTAOLC | The topology agent has written an LUcollection response string in an output buffer. The string is greater than the size of the output buffer and has overlayed other storage. | Mini |
| ISTTAC19 | ISTTAOLI | The topology agent has written an logicalUnitIndex response string in an output buffer. The string is greater than the size of the output buffer and has overlayed other storage. | Mini |
| ISTTAC20 | ISTTAONO | The topology agent, in processing notifications, has written data beyond the storage allocated for it. | Mini |
| ISTTAC21 | ISTTAOLG | The topology agent has written a transmission group distinguished name string to a data area that is smaller than the string. | Mini |
| ISTTAC22 | ISTTAOLT | The topology agent received a SNAlocaltopology message from configuration services containing an object block specifying an unknown object type. | Mini |
| ISTTAC23 | ISTTAOLC | The topology agent received an LUcollection message from configuration services containing an object block specifying an unknown or inappropriate object type. | Mini |
| ISTTAC24 | ISTTAOLI | The topology agent received a logicalUnitIndex message from configuration services containing an object block specifying an unknown or inappropriate object type. | Mini |
| ISTTAC25 | ISTTACA2 | The topology agent definitionGroupName attribute builder encountered an unknown type of major node. | Mini |
| ISTTAC26 | ISTTAOPU | The topology agent received a snapshot update data message from configuration services or topology and routing services containing an unrecognized return code. | Mini |
| ISTTAC27 | ISTTAOPI | The topology agent received a snapshot initial data message from configuration services or topology and routing services containing an unrecognized return code. | Mini |

*Table 35. FFST probes for the VTAM topology agent (ISTTACxx) (continued)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| ISTTAC28 | ISTTAOSN | The topology agent received a SNAnetwork message from configuration services containing an object block specifying an unknown or inappropriate object type. | Mini |
| ISTTAC29 | ISTTAISE | The topology agent has written a SET response string in an output buffer. The string is greater than the size of the output buffer and has overlayed other storage. | Mini |
| ISTTAC30 | ISTTAOSS | The topology agent received a synchronous return code from the CMIP services API indicating a permanent error has been encountered. | Mini |
| ISTTAC31 | ISTTAIPS | The topology agent received a CMIP SET message with a syntax error. | Mini |
| ISTTAC32 | ISTTAOUQ | The topology agent received a snapshot update message from configuration services containing an object block specifying an unknown object type. | Mini |
| ISTTAC33 | ISTTAOUQ | A message routed to the module for the topology agent merge queue controller contained an operation code that is not supported by that module. | Mini |

**FFST probes**

# Appendix D. Communications storage manager (CSM) FFST probes

## CSM FFST probe naming conventions

The following table describes the naming conventions for the FFST probe name. **IVTSMC01** is used as an example.

*Table 36. CSM FFST probe naming conventions*

| Characters | Example | Description |
|---|---|---|
| 1,2,3 | IVT | These characters represent the product identifier. For CSM probes, these characters are IVT. |
| 4, 5 | SM | These characters represent the CSM component identifier. In this example, SM is the component identifier for the Storage Manager component. |
| 6 | C | For CSM probes, this character is usually C. |
| 7, 8 | 01 | These characters represent the probe identification number. This number is not duplicated. |

## CSM probes

*Table 37. FFST probes for communications storage manager (IVTSMCxx)*

| Probe name | Module | Description | Dump type |
|---|---|---|---|
| IVTSMC03 | IVTSMCCP | Buffer size (BUFSIZE) or buffer source (BUFSOURCE) specification on a CREATE_POOL request was not valid. | Mini |
| IVTSMC04 | IVTSMCCP | SRB schedule of IVTSMCEX failed for a CREATE_POOL request. | Full |
| IVTSMC05 | IVTSMCGF | A storage error occurred. | Full |
| IVTSMC06 | IVTSMCFB | SRB schedule of IVTSMCEX failed during pool contraction. | Full |
| IVTSMC07 | IVTSMCEX | Cross-memory post failed. | Full |
| IVTSMC08 | IVTSMCFC | SRB schedule of IVTSMCEX failed during pool contraction. | Full |
| IVTSMC09 | IVTSMCFX | SRB schedule of IVTSMCSD failed. | Full |
| IVTSMC0A | IVTSMCBF | SRB schedule of IVTSMCSD failed. | Full |
| IVTSMC0B | IVTSMCFR | SRB schedule of IVTSMCSD failed. | Full |
| IVTSMC0C | IVTSMCWL | SRB schedule of IVTSMCSD failed. | Full |
| IVTSMC0D | IVTSMCFX | Post failed. | Full |
| IVTSMC0E | IVTSMCFR | Post failed. | Full |
| IVTSMC0F | IVTSMCFC | SRB schedule of IVTSMCSD failed. | Full |
| IVTSMC10 | IVTSMCGB | The schedule of the SRB for the side routine failed in IVTSMCFB. | Full |

**CSM probes**

# Appendix E. Architectural specifications

This appendix lists documents that provide architectural specifications for the SNA Protocol.

The APPN Implementers' Workshop (AIW) architecture documentation includes the following architectural specifications for SNA APPN and HPR:

- APPN Architecture Reference (SG30-3422-04)
- APPN Branch Extender Architecture Reference Version 1.1
- APPN Dependent LU Requester Architecture Reference Version 1.5
- APPN Extended Border Node Architecture Reference Version 1.0
- APPN High Performance Routing Architecture Reference Version 4.0
- SNA Formats (GA27-3136-20)
- SNA Technical Overview (GC30-3073-04)

For more information, refer to the AIW documentation page at http://www.networking.ibm.com/app/aiwdoc.htm.

The following RFC also contains SNA architectural specifications:

- RFC 2353 *APPN/HPR in IP Networks APPN Implementers' Workshop Closed Pages Document*

RFCs can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Many RFCs are available online. Hardcopies of all RFCs are available from the NIC, either individually or by subscription. Online copies are available using FTP from the NIC at http://www.rfc-editor.org/rfc.html.

Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

where:

- *nnnn* is the RFC number.
- TXT is the text format.
- PS is the postscript format.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC` *nnnn* for text versions or a subject line of `RFC` *nnnn*`.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil`.

**473**

# Appendix F. Information APARs and technotes

This appendix lists information APARs for IP and SNA documents.

**Note:**

1. Information APARs contain updates to previous editions of the documents listed in Table 38 and Table 39 on page 476. Documents updated for V1R9 are complete except for the updates contained in the information APARs that might be issued after V1R9 documents went to press.

2. Information APARs are predefined for z/OS V1R9 Communications Server and might not contain updates.

3. Information APARs for z/OS documents are in the document called *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS.

## Information APARs for IP documents

Table 38 lists information APARs for V1R6 IP documents. For releases V1R7 and later, updates are available as technotes, which can be found at http://www.ibm.com/support/docview.wss?uid=swg21178966.

*Table 38. IP information APARs for z/OS Communications Server*

| Title | Information APAR for V1R6 |
|---|---|
| New Function Summary (both IP and SNA) | II13824 |
| Quick Reference (both IP and SNA) | II13831 |
| IP and SNA Codes | II13842 |
| IP Sockets API Guide | II13844 |
| IP Configuration Guide | II13826 |
| IP Configuration Reference | II13827 |
| IP Diagnosis | II13836 |
| IP Messages Volume 1 | II13838 |
| IP Messages Volume 2 | II13839 |
| IP Messages Volume 3 | II13840 |
| IP Messages Volume 4 | II13841 |
| IPv6 Network and Application Design Guide | II13825 |
| IP Programmer's Guide and Reference | II13843 |
| IP User's Guide and Commands | II13832 |
| IP System Admininstrator's Commands | II13833 |

# Information APARs for SNA documents

Table 39 lists information APARs for V1R6 SNA documents. For releases V1R7 and later, updates are available as technotes, which can be found at http://www.ibm.com/support/docview.wss?uid=swg21178966.

*Table 39. SNA information APARs for z/OS Communications Server*

| Title | Information APAR for V1R6 |
|---|---|
| New Function Summary (both IP and SNA) | II13824 |
| Quick Reference (both IP and SNA) | II13831 |
| IP and SNA Codes | II13842 |
| SNA Customization | II13857 |
| SNA Diagnosis, Vol. 1: Techniques and Procedures | II13852 |
| SNA Diagnosis, Vol. 2: FFST Dumps and the VIT | II13853 |
| SNA Messages | II13854 |
| SNA Network Implementation Guide | II13849 |
| SNA Operation | II13851 |
| SNA Programming | II13858 |
| SNA Resource Definition Reference | II13850 |
| SNA Data Areas Volume 1 | II13855 |
| SNA Data Areas Volume 2 | II13856 |

# Other information APARs

Table 40 lists information APARs not related to documents.

*Table 40. Non-document information APARs*

| Content | Number |
|---|---|
| Index to APARs that list recommended VTAM maintenance | II11220 |
| Index to APARs that list trace and dump requests for VTAM problems | II13202 |
| Index of Communication Server IP information APARs | II12028 |
| Collecting TCPIP CTRACEs | II12014 |
| CSM for VTAM | II13442 |
| CSM for TCP/IP | II13951 |
| DLUR/DLUS | II12986, II13456, and II13783 |
| Documentation required for FTP server problems | II12925 |
| Documentation required for OSA/2, OSA Express and OSA QDIO | II13016 |
| DNS — common problems and solutions | II13453 |
| Enterprise Extender | II12223 |
| FTP client and FTP server TLS support | II13516 |
| FTP problems | II12079 |
| FTPing doc to z/OS Ssupport | II12030 |
| Generic resources | II10986 |
| HPR | II10953 |

*Table 40. Non-document information APARs  (continued)*

| Content | Number |
|---|---|
| iQDIO | II13142 |
| LPR problems | II12022 |
| MNPS | II10370 |
| MPC and CTC | II01501 |
| NCPROUTE problems | II12025 |
| OMPROUTE | II12026 |
| PASCAL API | II11814 |
| Performance | II11710<br>II11711<br>II11712 |
| Resolver | II13398<br>II13399<br>II13452 |
| Socket API | II11996<br>II12020 |
| SMTP problems | II12023 |
| SNMP | II13477<br>II13478 |
| SYSLOGD howto | II12021 |
| TCPIP connection states | II12449 |
| TN3270E Telnet server | II11574<br>II13135 |
| TN3270E Telnet server SSL common problems | II13369 |

# Appendix G. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

## Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

## Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

## z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

www.ibm.com/servers/eserver/zseries/zos/bkserv/

# Notices

IBM may not offer all of the products, services, or features discussed in this document. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application

programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California. Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Portions of this publication relating to RPC are Copyright © Sun Microsystems, Inc., 1988, 1989.

Some portions of this publication relating to X Window System** are Copyright © 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, and the Massachusetts Institute Of Technology, Cambridge, Massachusetts. All Rights Reserved.

Some portions of this publication relating to X Window System are Copyright © 1986, 1987, 1988 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute the M.I.T., Digital Equipment Corporation, and Hewlett-Packard Corporation portions of this software and its documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T., Digital, and Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T., Digital, and Hewlett-Packard make no representation about the suitability of this software for any purpose. It is provided ″as is″ without express or implied warranty.

Copyright © 1983, 1995-1997 Eric P. Allman

Copyright © 1988, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software program contains code, and/or derivatives or modifications of code originating from the software program "Popper." Popper is Copyright ©1989-1991 The Regents of the University of California, All Rights Reserved. Popper was created by Austin Shelton, Information Systems and Technology, University of California, Berkeley.

Permission from the Regents of the University of California to use, copy, modify, and distribute the "Popper" software contained herein for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies. HOWEVER, ADDITIONAL PERMISSIONS MAY BE NECESSARY FROM OTHER PERSONS OR ENTITIES, TO USE DERIVATIVES OR MODIFICATIONS OF POPPER.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THE POPPER SOFTWARE, OR ITS DERIVATIVES OR MODIFICATIONS, AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE POPPER SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Copyright © 1983 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of California, Berkeley. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1991, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright © 1990 by the Massachusetts Institute of Technology

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore

if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original M.I.T. software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided ″as is″ without express or implied warranty.

Copyright © 1998 by the FundsXpress, INC. All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided ″as is″ without express or implied warranty.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED ″AS IS″ AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be

given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)". The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related.

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include acknowledgement:

   "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

This product includes cryptographic software written by Eric Young.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats.

## Programming interface information

This publication documents information NOT intended to be used as Programming Interfaces of z/OS Communications Server.

# Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

| | |
|---|---|
| Advanced Peer-to-Peer Networking | NetView |
| AFS | Network Station |
| AD/Cycle | Nways |
| AIX | OfficeVision |
| AIX/ESA | OS/2 |
| AnyNet | OS/390 |
| APL2 | Parallel Sysplex |
| AS/400 | PROFS |
| BookManager | pSeries |
| C/370 | RACF |
| CICS | Redbooks |
| CICS/ESA | RETAIN |
| C Set ++ | REXX |
| Common User Access | RISC System/6000 |
| CUA | RMF |
| DB2 | RS/6000 |
| DFSMS | S/370 |
| DFSMSdfp | S/390 |
| DFSMShsm | S/390 Parallel Enterprise Server |
| DPI | SAA |
| ESCON | SecureWay |
| eServer | SET |
| ES/9000 | SiteCheck |
| FFST | SP |
| FICON | System/360 |
| First Failure Support Technology | System/370 |
| GDDM | System/390 |
| IBM | System z |
| ibm.com | System z9 |
| IBMLink | Tivoli |
| IMS | Tivoli Enterprise Console |
| IMS/ESA | VM/ESA |
| HiperSockets | VSE/ESA |
| Language Environment | VTAM |
| Micro Channel | WebSphere |
| Multiprise | z9 |
| MVS | z/Architecture |
| MVS/DFP | z/OS |
| MVS/ESA | z/VM |
| MVS/SP | zSeries |
| | 400 |

The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

PostScript is a registered trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

# Bibliography

## z/OS Communications Server information

This section contains descriptions of the documents in the z/OS Communications Server library.

z/OS Communications Server documentation is available:
- Online at the z/OS Internet Library web page at http://www.ibm.com/servers/eserver/zseries/zos/bkserv
- In softcopy on CD-ROM collections. See "Softcopy information" on page xx.

## z/OS Communications Server library

z/OS Communications Server documents are available on the CD-ROM accompanying z/OS (SK3T-4269 or SK3T-4307). Unlicensed documents can be viewed at the z/OS Internet library site.

Updates to documents are available on RETAIN and in information APARs (info APARs). See Appendix F, "Information APARs and technotes," on page 475 for a list of the documents and the info APARs associated with them.

Info APARs for z/OS documents are in the document called *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation* which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/ BOOKS/ZIDOCMST/CCONTENTS.

### Planning

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: New Function Summary* | GC31-8771 | This document is intended to help you plan for new IP for SNA function, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions. |
| *z/OS Communications Server: IPv6 Network and Application Design Guide* | SC31-8885 | This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server's support of IPv6, coexistence with IPv4, and migration issues. |

### Resource definition, configuration, and tuning

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: IP Configuration Guide* | SC31-8775 | This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Reference*. |

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: IP Configuration Reference* | SC31-8776 | This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Guide*. The information in this document includes:<br>• TCP/IP configuration data sets<br>• Configuration statements<br>• Translation tables<br>• SMF records<br>• Protocol number and port assignments |
| *z/OS Communications Server: SNA Network Implementation Guide* | SC31-8777 | This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the *z/OS Communications Server: SNA Resource Definition Reference*. |
| *z/OS Communications Server: SNA Resource Definition Reference* | SC31-8778 | This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document in conjunction with the *z/OS Communications Server: SNA Network Implementation Guide*. |
| *z/OS Communications Server: SNA Resource Definition Samples* | SC31-8836 | This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions. |
| *z/OS Communications Server: IP Network Print Facility* | SC31-8833 | This document is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services. |

## Operation

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: IP User's Guide and Commands* | SC31-8780 | This document describes how to use TCP/IP applications. It contains requests that allow a user to log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users. |
| *z/OS Communications Server: IP System Administrator's Commands* | SC31-8781 | This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process. |
| *z/OS Communications Server: SNA Operation* | SC31-8779 | This document serves as a reference for programmers and operators requiring detailed information about specific operator commands. |
| *z/OS Communications Server: Quick Reference* | SX75-0124 | This document contains essential information about SNA and IP commands. |

# Customization

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: SNA Customization* | SC31-6854 | This document enables you to customize SNA, and includes the following:<br>• Communication network management (CNM) routing table<br>• Logon-interpret routine requirements<br>• Logon manager installation-wide exit routine for the CLU search exit<br>• TSO/SNA installation-wide exit routines<br>• SNA installation-wide exit routines |

# Writing application programs

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference* | SC31-8788 | This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP. |
| *z/OS Communications Server: IP CICS Sockets Guide* | SC31-8807 | This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS® using z/OS TCP/IP. |
| *z/OS Communications Server: IP IMS Sockets Guide* | SC31-8830 | This document is for programmers who want application programs that use the IMS™ TCP/IP application development services provided by IBM's TCP/IP Services. |
| *z/OS Communications Server: IP Programmer's Guide and Reference* | SC31-8787 | This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended. |
| *z/OS Communications Server: SNA Programming* | SC31-8829 | This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain. |
| *z/OS Communications Server: SNA Programmer's LU 6.2 Guide* | SC31-8811 | This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.) |
| *z/OS Communications Server: SNA Programmer's LU 6.2 Reference* | SC31-8810 | This document provides reference material for the SNA LU 6.2 programming interface for host application programs. |
| *z/OS Communications Server: CSM Guide* | SC31-8808 | This document describes how applications use the communications storage manager. |

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: CMIP Services and Topology Agent Guide* | SC31-8828 | This document describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The document provides guide and reference information about CMIP services and the SNA topology agent. |

## Diagnosis

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: IP Diagnosis Guide* | GC31-8782 | This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center. |
| *z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures* and *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT* | GC31-6850 GC31-6851 | These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation. |
| *z/OS Communications Server: SNA Data Areas Volume 1* and *z/OS Communications Server: SNA Data Areas Volume 2* | GC31-6852 GC31-6853 | These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA. |

## Messages and codes

| Title | Number | Description |
|---|---|---|
| *z/OS Communications Server: SNA Messages* | SC31-8790 | This document describes the ELM, IKT, IST, IUT, IVT, and USS messages. Other information in this document includes: <br> • Command and RU types in SNA messages <br> • Node and ID types in SNA messages <br> • Supplemental message-related information |
| *z/OS Communications Server: IP Messages Volume 1 (EZA)* | SC31-8783 | This volume contains TCP/IP messages beginning with EZA. |
| *z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)* | SC31-8784 | This volume contains TCP/IP messages beginning with EZB or EZD. |
| *z/OS Communications Server: IP Messages Volume 3 (EZY)* | SC31-8785 | This volume contains TCP/IP messages beginning with EZY. |
| *z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)* | SC31-8786 | This volume contains TCP/IP messages beginning with EZZ and SNM. |
| *z/OS Communications Server: IP and SNA Codes* | SC31-8791 | This document describes codes and other information that appear in z/OS Communications Server messages. |

# Index

# Special characters

\*\*\*\* (FFST) trace record   32

# A

abend (abnormal end) trace records
  abend RELSTORE   86
  ABND SNAP   35
ABND trace record   35
ACA1 trace record   36
ACA2 trace record   38
ACA3 trace record   38
accessibility   479
ACI1 trace record   36
ACI2 trace record   38
ACI3 trace record   38
ACP1 trace record   39
ACP2 trace record   41
ACR1 trace record   39
ACR2 trace record   41
ACRC trace record   43
ACSN trace record   44
ACU1 trace record   45
ACU2 trace record   45
ADE trace record   46
AFSM trace record   48
AI1 trace record   49
AI2 trace record   51
AI3 trace record   51
ALS trace record   52
AP trace record   54
API option
  summary   17
  VIT trace records created
    AI1   49
    AI2   51
    AI3   51
    IO1   224
    IO2 (MVS)   226
    IO3 (MVS)   226
    RE   308
    UE   33, 367, 368
    UE1   369
    UE2   370
    UP   372
APPC option
  summary   17
  VIT trace records created
    ACA1   36
    ACA2   38
    ACA3   38
    ACI1   36
    ACI2   38
    ACI3   38
    ACP1   39
    ACP2   41
    ACR1   39
    ACR2   41
    ACU1   45
    ACU2   45

APPC option *(continued)*
  VIT trace records created *(continued)*
    RACR   303
    REML   311
    REMQ   311
    USI   373
    USI2   375
    USO   373
    USO2   375
    UVI   376
    UVO   376
APPC sense code   7
ARB trace record   55
AREL trace record   86
ARP trace record   87
ARP2 trace record   88
ARQ trace record   89
ARQ2 trace record   91
ARS trace record   87
ARS2 trace record   88
ASN2 trace record   93
ASNB trace record   92
ATCASLIP   7
ATSK trace record   94
ATT (attention) trace record   95
authorized IO trace record   49
AXIT trace record   98

# B

BSPx trace record   96
BTSK trace record   99
BUFF trace record   100
buffer pool expansion, EXPN trace   189

# C

CCI or CCO trace record
  NCSPL   102
  not RUPE or NCSPL   104
  RUPE   101
CDHx trace record   105
CDN trace record   105
CDN2 trace record   106
CFA2 trace record   108
CFAL trace record   107
CFC2 trace record   111
CFCE trace record   109
CFCN trace record   110
CFDS trace record   111, 112
CFEE trace record   113
CFER trace record   118
CFFC trace record   118
CFL2 trace record   121, 123, 124, 125, 127
CFL3 trace record   122, 125, 126
CFLS trace record   119
CFNF trace record   127
CFP2 trace record   129
CFPG trace record   128
CFRB trace record   130

**495**

# S

# T

# Communicating Your Comments to IBM

If you especially like or dislike anything about this document, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Please send your comments to us in either of the following ways:
- If you prefer to send comments by FAX, use this number: 1+919-254-1258
- If you prefer to send comments electronically, use this address:
  – comsvrcf@us.ibm.com
- If you prefer to send comments by post, use this address:

  ```
  International Business Machines Corporation
  Attn: z/OS Communications Server Information Development
  P.O. Box 12195, 3039 Cornwallis Road
  Department AKCA, Building 501
  Research Triangle Park, North Carolina 27709-2195
  ```

Make sure to include the following in your note:
- Title and publication number of this document
- Page number or topic to which your comment applies.

**IBM** ®

Program Number:  5694–A01

Printed in USA

Spine information:

z/OS Communications Server

z/OS V1R9.0 Comm Svr: SNA Diagnosis Vol 2, FFST Dumps and the VIT

Version 1
Release 9

IBM