Comm-Pro Associates
Host Network Access Support

HNAS

Console Subsystem Operations Guide & Trace Formats

V2R4M0

Please see next page for important information concerning your Host NAS distribution materials.

This file revised November 8, 2012 9:30 am, includes maintenance thru 2400111 5.

General Information

Comm-Pro Associates is the designer and developer of the X.25 Host Network Access Support host resident Program Product (Commonly referred to as **Host NAS** or **HNAS**). The Host NAS product provides support for X.25 connectivity over router based networks using IBM's X.25 through TCP/IP (**XTP**) or Cisco's X.25 over TCP/IP (**XOT**) transport protocols. The HNAS implementation was designed to avoid application changes by providing a transparent migration from IBM 3745 NSPI based communication controllers to the HNAS router based solution. The product offering provides a robust suite of non-SNA NPSI type support. Please refer to Chapter 1 (Introduction) for a complete list of products and Applications supported.

Contact Information

Phone: (661) 284-3650

Fax: n/a

E-mail: support@comm-pro.com

FTP: ftp.comm-pro4ftp.com
(Userid required, available upon request)

WWW: www.comm-pro.com

Shipping Address: 25852 McBean Parkway #611
Santa Clarita, CA USA 91355-3705

For additional documentation and up-to-date information, please refer to member @README in the Comm-Pro distribution macro library. See our WEB site for the latest information.

Important Notes

1) Please refer to the optional README/@README file included with the HNAS product distribution media (separate file or HNASMAC macro member) for additional product information and documentation not included in this manual. Additional information can also be located on our web site (Please refer to Contact Information section for contact details).

Preface

Comm-Pro Biography

Comm-Pro Associates, Inc. was formed in 1973 to provide software packages designed to enhance the performance of early IBM 3705 communications controllers. IBM sold Comm-Pro's PP04 software package as the PEP Extended Features PRPQ. Comm-Pro is a privately held corporation.

Comm-Pro's X.25 Network Access Support (NAS) provided X.25 access to EP and NCP host applications and provided expanded support over what was provided by NPSI. Over the years Comm-Pro has worked extensively with IBM on special projects and custom environments. This product was introduced in 1976 and continues to be in use today at a limited number of customer sites. That's over 30 years of service.

HNAS, which allows X25 links to communicate with host programs without the need for a 37xx controller, was first implemented for IBM routers employing the XTP (X25 over TCP/IP) protocol in 1998. IBM specifically selected Comm-Pro to develop the software and much of the initial testing was done at IBM Raleigh. HNAS was developed for Cisco routers employing the XOT protocol in 1999. Due to changes in IBM's business strategy the router line was dropped. As a result, HNAS now is primarily used with Cisco routers and German AGIS Bin-Tec routers.

Special Notices

This book is furnished as is. Comm-Pro assumes no responsibility for the use of the functions described in this book in any manner.

The Host NAS licensed program described in this documentation and all license material available for it are provided by Comm-Pro under terms of the Software Use Agreement provided by Comm-Pro or it's Business Partner's.

Trademarks

X.25 Host NAS is fully year 2000 compliant.

This page left intentionally blank.

Related Publications

Related publications, providing reference material for this product are:

RFCs
- RFC1613 - XOT (X25 Over TCP)

Comm-Pro Associates
- Host NAS Configuration Guide and Reference
- Host NAS Messages and Codes Debugging Guide
- Host NAS Console Subsystem Operations Guide
- Host NAS Console Subsystem Operations Guide & Trace Formats
- Host NAS Master Index (Including Master Revision Index)

Please refer to the Documentation Overview section for additional information on the HNAS documentation organization and edistribution locations.

Cisco Systems

Keywords - XOT (X.25 over TCP/IP, formerly tunneling), X.25 Switching and X.25 LAPB)
- Cisco IOS Configuration Fundamentals, Network Protocols and various modules
- Cisco IOS Wide-Area Networking Configuration Guide - X.25 and LAPB (78-11751-01)
- Cisco IOS Wide-Area Networking Command Reference - X.25 and LAPB (78-011752-01)
- Cisco IOS Software Command Summary
- Cisco IOS Software Error Messages
- Cisco Debug Command Reference (Use with Internetwork Troubleshooting Guide)

Cisco Connection online documentation is available online at the following Web Site link:

www.cisco.com/univercd/home/home.htm

IBM Corporation

Keywords - XTP (X.25 through TCP/IP), X.25 MAS and X.25 LAPB)
- IBM Communication Controller Migration Guide (/redbooks/SG246298.html)
- IBM NCP and NPSI - X.25 Planning and Installation (SC30-3470-nn)
- IBM NCP and NPSI - X.25 Diagnosis, Customization and Tuning (LY30-5610-nn)
- IBM - IP Application Programming Interface Guide (SC31-8788)
  (TCP/IP Stack information, including the list of TCP/IP Stack **ERRNO** return codes
- IBM - Access Integration Services - Software User's Guide (SC30-3988/SC30-3998)
  (NWAYS Multiprotocol Access Services - IBM 22nn)

Several IBM documentation manuals relating to **ACF/NCP, ACF/SSP, ACF/VTAM, EP** and **NPSI** are available for viewing or downloading at the following IBM Web Site link:

www.networking.ibm.com/375/public.html

This page left intentionally blank.

Emergency Support Information

Comm-Pro Associates, Inc. HNAS emergency technical support is available by contacting our office, phone attendants are available 24 hours a day, 7 days a week. Our technical support representatives are usually available to provide immediate support during our standard business hours of 07:00 AM to 04:30 PM Monday through Friday. There are times during our **standard** business hours when phone calls will initially be handled by our non-technical phone attendants. In these cases, calls will be handled the same as our non-standard hours support.

During **non-standard** business hours, our phone attendant will gather any appropriate customer information for call back. The attendant will then attempt to locate a technical support representative to assist you. Your phone call will either be directly connected to a technician or receive a call back with-in a reasonable time frame.

We recommend that all customers provide problem descriptions **via E-mail** as well as trace or dump (ABEND) diagnostics via E-mail or ftp so that we can begin work on the problem as soon as possible. It's a good idea to always provide a copy of the CDF (configuration data file) and the product level information (see DNAS display output which is provided at the beginning of all HNAS sysprint output).

You may also obtain emergency support by e-mail notification. Under emergency situations your e-mail transmission should always be followed up with a phone call if you haven't received an e-mail response with-in a reasonable time frame. This is primarily because our e-mail services are primarily only monitored during our standard office hours. Please refer to our contact page for contact numbers and E-mail addresses.

International users should contact their Comm-Pro Business Partners for first level support, if applicable.

General Support Information

General product support is provided to all licensed HNAS users. Responses to customer inquiries are typically provided with-in 24 hours during normal business hours.

As indicated under Emergency Support Information above, we recommend that all customers provide problem reporting (as well as support questions regarding configuration or run time issues) via E-mail.

Questions or problems reported via e-mail outside of our standard business hours will often be responded to on the day of the inquiry if received before 10:00 PM Pacific Time (GMT-8).

FAX / Facsimile - Direct service no longer provided

We no longer provide direct Fax / Facsimile service in our offices. E-mail and FTP are the primary methods supported for electronic document or image delivery.

Should there be a requirement to send a Fax from your organization to Comm-Pro there are a couple of potential options:

- Make arrangements with a fax-to-email service that can accept your fax and route via email to Comm-Pro at support@comm-pro.com.

- Call (during standard business hours) or e-mail a Comm-pro representative to see if a onsite fax session can be temporarily enabled with a temporary phone number, if available.

Note: Our old fax numbers 661/287-1646 and 661/291/2324 were retired and are no longer in service.

HNAS Documentation Overview

The HNAS product documentation manuals (currently 4 primary publications and a master index guide) are provided below with their respective sections listed for ease of information source retrieval and viewing. Supplemental booklet references (as applicable) are also provided in this section.

Documentation manuals for the HNAS products are available in PDF format for individual books (vrm_HNASBook|MsgCodes|Console|ConsTrc|MasterIndex_yyyy-mm-dd.pdf) and collectively in zip archive files (vrm_HNASBooks_yyyy-mm-dd.zip).

These manuals are primarily available on our FTP server (userid required) or can be obtained by contacting a HNAS marketing and support representative for an alternate form of PDF documentation delivery. The PDF files were designed for duplex printing although the content can be printed in simplex (one sided) mode.

All HNAS documentation manuals and books provide the revision date on the bottom left corner of the header page "**This file revised Month, day (dd), year (yyyy) time (hh:mm) am|pm"**. As of February 15, 2006, "**,includes maintenance thru 2400nnn.**" is also provided in the header page to reflect the maintenance level that the documentation level reflects.

The documentation manuals are downward compatible with older HNAS releases. Every effort is made to identify and label new features or changes at the HNAS vrm level that the change was introduced. As we add APARs fixes and enhancement content, we denote the APAR numbers associated with the new parameters, alert messages and content whenever possible.

HNAS APAR Maintenance Level included in this Documentation:

2400nnn	APAR maintenance level included in this documentation series. Refer to HNAS book cover page 'includes maintenance thru 2400nnn' text.

HNAS Guide and Reference:

-Book File-	240_HNASBook_yyyy-mm-dd.pdf - PDF Format
Prefix	(Prefix) General Information, Contact Information and important Notes. *
Preface	(Preface) Comm-Pro Biography, Special Notices, Trademarks, Related Publications, Emergency Support and General Support. *
DocOView	(Documentation Overview) Descriptive list of HNAS Documentation manuals (books) and Sections. *
Chapter 1	(Introduction) describes the features of the HNAS software.

HNAS Guide and Reference:

Chapter 2	(Installation, Activation and Runtime Guide) describes the procedures used to install the Comm-Pro software from its distribution medium and how to generate and execute an HNAS load module program.
Chapter 3	(Configuration Guide) describes the operational characteristics of Comm-Pro's HNAS software and illustrates how to use configuration definition statements to define HNAS resources.
Chapter 4	(Configuration Reference) describes the configuration definition statements and parameters that are used to define HNAS resources.
Chapter 5	(Migration Reference) describes the configuration operands and run time functions that have changed in this release of HNAS. **Note**: It is important that you review this section prior to refreshing/upgrading from an older HNAS release.
Chapter 6	(Maintenance and APAR Summaries) provides information on maintenance types, installation and APAR (PTF) maintenance memo formats. Memo's are available on the HNAS maintenance Web site FTP Server or via E-mail subscription.
Appendix A	(X.3 PAD Parameters) describes X.3 PAD parameters.
Appendix B	(Configuration Examples) provides an example HNAS configuration data file and the resulting SYSPRINT log files.
Appendix C	(Router Checklist Overview) currently provides a basic overview for defining XOT and X.25 support in a Cisco router for HNAS connectivity. Also describes some of the Cisco diagnostic show and debug commands.
Appendix D	(Changes & New Features) provides an overview of new features provided in the current release as well as historical data for previous releases. In 220 and earlier releases of HNAS this content was provided in the Preface section.
Glossary	(Glossary of Terms) currently provides a reference list and some brief definitions for terms, abbreviation and acronyms that may be used in the HNAS documentation manuals, ftp or web page content.

HNAS Messages and Codes Debugging Guide:

-Book File-	240_MsgCodes_yyyy-mm-dd.pdf - PDF Format
CnfgMsgs	(Configuration Messages) provides information for HNAS configuration messages (Information, Default, Warning, Error, etc.) that can be encountered during HNAS initialization when processing the Configuration Data File (CDF).

HNAS Messages and Codes Debugging Guide:

AlrtMsgs	(Alert Messages) provides information for HNAS alert messages (Info, Warning, Error and Severe) that can be encountered during HNAS activation (after the CDF scan) and during “run time” operation.
BindfCodes	(BIND Failure User Sense Codes) describes reason for BIND failures.
TcpipErrno	(TCP/IP Error Numbers ERRNO) describes reason for TCPIP Errors.
PvcssCodes	(PVC Setup Status Codes RFC-1613) describes PVC Setup Ending Status.
RstCodes	(X.25 Reset Cause and Diagnostic Codes) describes the X.25 Reset Cause and Diagnostic codes that are present in the HNAS environment.
ClrCodes	(X.25 Clear Cause and Diagnostic Codes) describes the X.25 Clear Cause and Diagnostic codes that are present in the HNAS environment. In 230 Extended Diagnostic reason codes were added to the respective clear code entries to further define the cause of the event.
CisMsgs	(Cisco Messages Relating to HNAS Events) describes common Cisco codes in relationship to HNAS events.
SysAbnd	(System Abend Codes - Messages Relating to HNAS Events)
HaltMsgs	(HNAS HALT/NASHALT Messages Relating to HNAS ABEND Events)
ConsMsgs	(Console Command Error Messages) provides diagnostic error messages for some HNAS console subsystem commands.

HNAS Console Subsystem Operations Guide:

-Book File-	240_Console_yyyy-mm-dd.pdf - PDF Format
Console	(Console Subsystem) This document contains the same Console section content as the primary Console Subsystem Operations Guide but does not contains the Trace Entry Formats section. This section was designed for users who prefer to view or print the guide but don’t require use of the estimated 70 pages of Trace Entry Formats.
ConsMsgs	(Console Command Error Messages) provides diagnostic error messages for some HNAS console subsystem commands.

HNAS Console Subsystem Operations Guide & Trace Formats:

-Book File-	240_ConsTrc_yyyy-mm-dd.pdf - PDF Format

HNAS Console Subsystem Operations Guide & Trace Formats:

Console	(Console Subsystem) includes the Console Users Guide that describes the operation of the HNAS console subsystem for local or optional remote consoles.
ConsMsgs	(Console Command Error Messages) provides diagnostic error messages for some HNAS console subsystem commands.
Trace	(Trace Entry Formats) this section provides HNAS trace table entry identifiers, layouts and descriptions of the various trace entries provided by HNAS.

HNAS Master Index - Index Entries for All HNAS Manuals:

-Book File-	240_MasterIndex_yyyy-mm-dd.pdf - PDF Format
Master-Index	(Master Indexes) This document contains the master index. The master index contains the combined book indexes for all of the above referenced HNAS manuals and guides. On 03-17-2004 the Master Revision Index section was removed to avoid confusion that some customers were encountering when searching through the index. This section is now available upon request. **Note**: The master revision index was designed for documentation change control and doesn't contain content suitable for indexing.

HNASBooks in Pkware ZIP Format - All HNAS Manuals:

-All Books-	240_HNASBooks_yyyy-mm-dd.zip - ZIP archive of PDF books
HNASBooks	This zip file contains a collection of HNAS documentation manuals for single file transfer download operation. File CONSTRC is not included is this set, download separately.

***** - Denoted sections available in all documentation manuals.

All HNAS manuals and guides include Prefix sections (General Information, Important Notes), Preface sections (Special Notices, Trademarks, Related Publications), Documentation Overview, Table of Contents and Index sections. See 'Vendor Reference' index entries for additional vendor documentation references.

In 220, The Revision Index was removed from the individual manuals on 07/11/2003 in an effort to eliminate confusion. The Revision Index is still available for viewing in the Master Index manual.

HNAS Documentation Format

Documentation manuals for the HNAS products are available in PDF book format for individual books ***(vrm_book-name_yyyy-mm-dd*.pdf)** and collectively in zip archive file format **(*vrm*_HNASBooks_*yyyy-mm-dd*.zip)**. The PDF files were designed for Adobe Reader viewing and duplex printing although the content can be printed in simplex (one sided) mode. Some documentation content is available in HTML format on our Web site.

HNAS Documentation Locations

HNAS documentation manuals are available for customer download (using registered HNAS FTP server userid/password) at the following FTP Site address:

ftp://ftp.comm-pro4ftp.com/

HNAS documentation information is available online at the following Web Site link:

www.comm-pro.com/hostnas/docs/docindx.htm

Alternate forms of documentation delivery (e-mail file attachment or physical media) can be arranged by contacting your HNAS marketing and support representative. For Comm-Pro directly supported customers, we suggest that you send an e-mail request to the following address with the text 'HNAS Documentation Request' in the subject field of the E-mail:

support@comm-pro.com

HNAS Documentation Maintenance

Every effort is made to provide accurate and up-to-date product documentation for our users. Please don't hesitate to contact us with any corrections or recommendations regarding any of our documentation content. We appreciate your input and efforts.

This page left intentionally blank.

Console/Trace Table of Contents

Console Subsystem

This page left intentionally blank.

Introduction

This chapter describes the operations that may be performed using the Host NAS Console Subsystem.

Note: *Before reading the description of each console command, it is highly recommended that you first review the sections entitled* ***Terminology*** *on page CONS-3,* ***Output Rules*** *on page CONS-13,* ***Input Rules*** *on page CONS-20,* ***Command Description Rules*** *on page CONS-20 and* ***Miscellaneous Rules*** *on page CONS-21.*

A console user may be local or remote. A **local console** is either the main z/OS, OS/390 or MVS System Console (**SYSCONS**) operator or a TSO user with operator console privileges. A **remote console** is an ITI (LLC5/Pad) end user session connected over a TCP/IP network via the XOT or XTP protocol.

Local console operators can request functions that remote console operators cannot. For example, local console operators can request HNAS shutdown using the 'Q' (QUIT command). The 'Q' command for remote consoles causes the remote console session to be terminated and an XTP or XOT Clear Request packet to be transmitted.

A local user (system's operator or a *privileged* TSO user) becomes an HNAS console operator as soon as HNAS is started. Local console operators communicate with HNAS via the modify command interface (when the USEMDFY start parameter is in effect) or via the WTOR interface (when the USEWTOR start parameter is in effect). In the latter case, HNAS always has a WTOR outstanding that can be answered by the local user.

Note: The **USEMDFY** start parameter is a default start parameter and need not be specified. If WTOR interface support is required, you must specify the USEWTOR start parameter.

Note: When the **USEMDFY** start parameter is in effect, the WTOR reply is replaced by Modify command input (**/F *jobname*,*command***).

A remote user becomes an HNAS console operator by establishing an LLC0 (PCNE) or LLC5 (PAD) session and selecting the APPLNAME entry whose name is 'CONSOLE'. See Chapter 4 of the HNAS Guide and Reference manual for a description of the SYSL operand of the REMOTE definition statement for examples of how the HNAS 'system select' function operates. Remote console access can be password protected so that the remote user must enter a configured password before console status is attained.

If all console control blocks are in use when a remote user requests console status, the remote call is cleared. Also, if the remote user fails to enter the configured password within 30-seconds or if password verification fails after three (3) attempts, the remote call is cleared.

When successful connection to the Console Subsystem is established (local or remote), the console operator is prompted for a console command (e.g., **`ENTER COMMAND:`**) using the text supplied by the **CONPRMT** operand on the BUILD definition statement.

Commands identified in this document as ***privileged*** will only be executed for privileged consoles. The local SYSCONS operator or a TSO user with system console access always has

privileged status. A remote console operator can attain *privileged status* only if the **RMT-CONP** start parameter or **RMTCONS PRIV** local console command has been specified and the console *password is entered backwards*. For more information, refer to the description of the **CONPSWD** operand of the BUILD definition statement in Chapter 4 of the HNAS Guide and Reference documentation.

Terminology

SYSCONS

Refers to the local systems console, MCS console or a TSO user with console authorization.

Alert and Alarm Messages

Unless noted otherwise, references to alert or alarm messages are treated identically.

LU and SLU

Unless noted otherwise, references to LU or SLU are treated identically.

PU and SPU

Unless noted otherwise, references to PU or SPU are treated identically. Normally used in discussion of QLLC (LLC3) resources.

PLU

PLU refers to the host application (CICS, IMS, TSO, etc.).

Global Actions

Refers to an action (tracing, statistics collection, etc.) that is system wide affecting **all like** resources with action flags set in the HNAS common workarea (XFNASWA).

For example, TRCBFR ALLON activates I/O buffer tracing for **all** PCEs with trace flags set in the WAOPT1 field in XFNASWA.

Local Actions

Refers to an action (tracing, statistics collection, etc.) that affects **a collection of like** resources with action flags set in the associated resource control blocks.

For example, TRCBFR ON activates I/O buffer tracing for the **specific** PCEs identified by the ID= or RNM= modifier with trace flags set in each identified PCE.

Prefix Command Modifiers

Refers to a command modifier entered **by itself** or on the **left side** of a command. A prefix command modifier sets a permanent modifier remembrance which is used for all subsequent commands that require the modifier and remains in effect until a new prefix command modifier value is entered. Multiple prefix command modifiers can be entered on the same input line before (to the left of) the first command. Prefix command modifiers can be overridden by suffix command modifiers (see below) for the duration of a single command only.

Suffix Command Modifiers

Refers to a command modifier entered on the **right side** of a command. A suffix command modifier sets a temporary modifier value which is used for the current command only. Multiple suffix command modifiers can be entered on the same input line after (to the right of) a command but before any command arguments. Suffix command modifiers are not remembered across commands.

Default Suffix Command Modifiers

When a temporary suffix (**right side**) command modifier is not entered, it's value is set to the corresponding permanent prefix (**left side**) command modifier. All commands that require modifiers echo the temporary values only.

Parameters, Arguments, Operands

Refers to a command variable. These terms are used interchangeably throughout this document.

Resource State Indications

Refers to the state a resource is in. These terms **active**, **ACT**, **online**, **ONLN** are used interchangeably throughout this document as are **inactive**, **INACT**, **IDLE**, **offline**, **OFLN**.

General notes on Terminology:

1) When a console command can control both a *global* and *local* action, the ALLON and ALLOFF follower is used to manipulate the *global* action while the ON and OFF follower is used to manipulate the *local* action.

2) For those console commands that can toggle the value established by a start parameter (*global* action), the ALLON|ALLOFF follower is used instead of the ON|OFF follower if the same command can manipulate both *global* and *local* actions. For example, the TRCLU

ALLON and TRCLU ALLOFF console commands turn on and off, respectively, the value set by the TRCLU start parameter.

Input Rules

Input Case

The Console Subsystem will recognize and process characters in upper or lower case. The case of alphabetic input is not significant.

End Character

Normal input is terminated by a carriage return character (remote console) or the ENTER key (local console).

Cancel Character

The BREAK key (remote console only) may be used to terminate output or cancel the current line of input.

Delete Character

A backspace character (CTL-H on a remote console) will delete the preceding input character (if any).

Parentheses ( )

are used to enclose a sequence of values that may be coded in one operand.

Commas ,

are used to separate a sequence of values that may be coded in a single operand. For example:

RNM=*rmtname* MRMT OPTIONS=PFXDCEADDR,REPDCEADDR

Blanks

are used to separate a sequence of operands if a command accepts multiple operands. For example:

RNM=*rmtname* DRMT OPTIONS SVC0 SVC5

Omitted Suboperands

in a list form operand are identified by two (2) consecutive commas (,,).

Omitted Values

in a multi-value suboperand are identified by two (2) consecutive slashes (//).

Quotes ‘’

are used to enclose a character string if that string can be confused with a keyword value for an operand or if the character string contains blank (space) characters.

Sublist

operands are those that can have a sequence of values each of which is separated by a comma and all of which are enclosed in parentheses. For example, PADPARM=(*pn*1,*pv*1,...,*pn*i,*pv*i)).

Input Record Length

The maximum size of an input line is 128 bytes. Backspace characters and the text they delete do not count in this limit. The HNAS console input parser does not allow a command to span input lines. All data for a command must be entered on the same line as the command.

Numeric Input

Most numeric input is in hexadecimal unless noted otherwise.

Command Separators

Commands must be separated from their parameters and other commands by at least one space character.

Parameter Separators

Parameters should be separated from other parameters for the same command by one or more space characters.

Examples:
```
PRNT CONS ON
TRCMCH DBK OFF
```

Input Parsing

The console processor parses input **left to right**.

Command Parameters

Command parameters like FMT2 are entered on the **right side** of a command (e.g., DPCE FMT2) and are in effect for the given command only.

Duplicated Parameters

If the same parameter is entered more than once for a command, an error message will be issued in response to operator input.

Example:

If you enter: **`TRCPCE ALLON ALLON`**

The following error message will issued because ALLON is entered more than once.

```
NASC512E PARAMETERS DUPLICATED, TRCPCE COMMAND ABORTED
```

Conflicting Parameters

If a command modifier is in conflict with a command parameter or if one parameter is in conflict with another, an error message will be issued in response to operator input.

Examples:

If you enter: **`DLU RNM=`*`spuname`*` TYPE=GATE`**

The following error message will be issued because GATE SLUs cannot be configured on an SPU. SPUs support QLLC SLUs only.

```
NASC311E RNM=spuname TYPE=SPU INVALID, DLU COMMAND ABORTED
```

If you enter: **`DLU RNM=`*`mchname`*` TYPE=QLLC`**

The following error message will issued because QLLC SLUs cannot be configured on an MCH. MCHs support PCNE, GATE and PAD SLUs only.

```
NASC311E RNM=mchname TYPE=MCH INVALID, DLU COMMAND ABORTED
```

If you enter: **`TRCPCE ALLON ALLOFF`**

The following error message will issued because ALLON and ALLOFF are mutually exclusive.

```
NASC511E PARAMETERS CONFLICT, TRCPCE COMMAND ABORTED
```

Multiple Parameters

Most console commands will accept multiple parameters (arguments) on the same command invocation. The parameters are collected and then acted upon.

For example, TRCLU *mchname* ON DBK MAXDATA is valid invocation of the TRCLU command. The ON, DBK and MAXDATA parameters are decoded and remembered. The order in which they are specified is arbitrary. However, the order in which they are processed is not. For commands like TRCLU, once their parameters are collected, the parameters are then processed in a given order as a function of the command. For these commands, their parameter processing order is listed in a 'Parameter Processing Matrix' as part of the command description.

Multiple Commands

The console processor will accept multiple commands in a line of input. However, the console processor will stop parsing the command string when a display command is encountered.

For example, if a console operator entered the command 'TRCVC ALLON TRCLU ALLON ID=3 DPCE TRCDISP ON', the console processor would execute the TRCVC ALLON command first, the TRCLU ALLON command second, the ID=3 modifier third and the DPCE command last. The TRCDISP ON command is not executed because the DPCE command generates display output which causes the console parser to terminate its processing.

Command Lists

Command lists provide another way to stack multiple commands for execution. Unlike the 'Multiple Commands' processing described above, command lists allow consecutive display commands to be scheduled for execution.

The **EXEC *ddname*** command allows you to specify a DDNAME that exists in the HNAS start job that points a file containing a list of console commands. The DDNAMEs that you provide in the HNAS start job are arbitrary: For example:

```
//CMDLIST1 DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST1),DISP=SHR
//CMDLIST2 DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST2),DISP=SHR
```

:

The **EXEC** (***command*1**,***command*2**,***command3***,...) command can also be entered on a single line without having to invoke a ***ddname***. Prior to the introduction of the EXEC command, only multiple non-display type commands could be entered and processed on a single line. Using the EXEC command, multiple display commands can be scheduled. For example, if EXEC (DNAS,DPARM) were entered, the DNAS and DPARM commands would be executed consecutively.

Command Modifiers

Command modifiers like ID= that are entered by themselves or on the **left side** of a command (e.g., ID=1-3 DPCE) become **permanent** and are remembered across commands. They remain in effect until they are changed. These modifiers are termed **prefix** modifiers.

Command modifiers that entered on the **right side** of a command (e.g., DPCE ID=4-5) are **temporary** and override the permanent command modifiers of the same name for the duration of the given command only. These modifiers are termed **suffix** modifiers.

Note: Prefix (**left side**) command modifiers become the current command modifiers for a command if suffix (**right side**) command modifiers are not entered. This can have unintended effects if you do not remember the value for a permanent command modifier that is used by a command. For this reason, you may wish to avoid using left side modifiers and always enter the modifiers that are required by a command on the right side.

Note: Permanent command modifiers can be displayed using the DPARM console command.

PCE name as a Suffix Command Modifier

For commands that operate on PCEs, you may specify a PCE name (***pcename***) or a static PCE alias (***pcestatic*** = **LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) as a suffix (**right side**) command modifier. For example, TRCDISP LCLCONS will start dispatcher tracing for the local console (SYSCONS).

pcename or ***pcestatic*** are now accepted as the first argument for the **DPCE**, **STATS**, **TRCBFR**, **TRCDATA**, **TRCDISP**, **TRCIO** and **TRCPCE** commands. LCLCONS, NASUTIL, PING, TIMER, XOTUTIL, XTPUTIL are the aliases, respectively, for the local console PCE (WACONPCE), the HNAS utility PCE (WANASPCE), the PING PCE (WAPNGPCE), the timer PCE (WATMRPCE), the XOT utility PCE (XOTUTIL) and the XTP utility PCE (XTPUTIL). *pcename* is the PCE name for a target PCE. PCE names can be displayed using the DPCE ID= RNM= command.

Notes: 1) *pcestatic* = LCLCONS|NASUTIL|PING|TIMER|XOTUTIL|XTPUTIL

Alternate names for the *pcestatic* argument are as follows:

```
WACONPCE = LCLCONS
WANASPCE = NASUTIL
WAPNGPCE = PING
WATMRPCE = TIMER
XOTUTIL  = XOTUTIL
XTPUTIL  = XTPUTIL
```

2) When the *pcestatic* argument is specified, it directs command processing to named static PCE.

3) When the *pcename* argument is specified, it directs command processing to the named PCE.

4) When *pcestatic* and *pcename* are omitted, the command operates off of the RNM= or ID= modifier. When the RNM= modifier is used, command processing is directed at the named REMOTE. When the ID= modifier is used, command processing is directed at the identified PCEs.

5) PCE names can be provided by issuing the DPCE ID= RNM= command.

Note: *pcename* and *pcestatic* argument support was introduced into 240 with Enhancement APAR 2400075.

Console PCE name as a Suffix Command Modifier

For commands that operate on console PCEs, you may specify a remote console PCE name (***rconname***) or the static console PCE alias (**LCLCONS**) as a suffix (**right side**) command modifier. For example, EXEC LCLCONS *ddname* when entered from a remote console will force the command list identified by *ddname* to be executed for the local console (SYSCONS).

rconname or **LCLCONS** are now accepted as the first argument for the **ALARM**, **EXEC**, **SCHEDULE**, **SMSG** and **TRCCONS** commands. LCLCONS is the alias for the local console PCE (WACONPCE). *rconname* is the PCE name for a target remote console. Console PCE names can be displayed using the DPCE ID= RNM= TYPE=CONS command.

Notes: 1) When the LCLCONS argument is specified, it directs command processing to the local console. When LCLCONS and *rconname* are omitted for the local console, LCLCONS is assumed.

2) When the *rconname* argument is specified, it directs command processing to the named remote console. When LCLCONS and *rconname* are omitted for a remote console, *rconname* is forced to the name of the remote console.

3) Console names can be provided by issuing the DPCE ID= RNM= TYPE=CONS command. For the ALARM, EXEC, SCHEDULE or SMSG commands, only consoles that are in the active state can be targeted by the *rconname* parameter (the TRCCONS command is excepted).

4) The LCLCONS and *rconname* arguments are valid for privilege consoles only when specified for the EXEC or SCHEDULE command. These commands are normally non-privileged while the ALARM command and all trace commands are privileged.

5) When a command is executed, the normal console command echo prefix is HNASCMD->. This prefix is changed when the command is the result of an EXEC command, SCHEDULE command, comes from another console or is the result of a TRAP. In these cases, the HNASCMD-> echo prefix is replaced with EXECCMD->, SCHDCMD->, PRXYCMD-> or TRAPCMD->, respectively, so that these commands can be identified in the SYSPRINT log file.

Note: *rconname* and LCLCONS argument support was introduced into 240 with Enhancement APAR 2400075.

Command Decode Failures

Input that cannot be decoded as a command will generate the following error message:

`NASC003E DECODE FAILURE FOR` *`badcmd`*`..., RE-ENTER`

This can occur if the console command *badcmd* contains a spelling or syntax error. This can also occur if a non-printable character was inadvertently entered. When a decode failure occurs, additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

Note: Many commands accept a resource name following the command. For example, DRMT *rmtname* SVC0. If *rmtname* is spelled incorrectly or if it is not defined in CDF on a REMOTE definition statement, the command may be rejected for the lack of a valid RNM= or ID= modifier which will cause the *rmtname* to be processed as the *badcmd*. This is because commands that are rejected for a missing or erroneous LUNM=, LNM= RNM=, or ID= modifier return control to the console input parser which will continue processing the current input record from the end of the rejected command. In the example above, this would be after DRMT which means that *rmtname* would be processed as if it were a new command resulting in the NASC003E message above.

Output Rules

Console Output Overview

Console output for unsolicited events (such as program operational status, alerts, traces and monitor entries) and for local or remote console solicited command output (display command output for non event or alert messages) are routed to SYSCONS (operator console), SYSPRINT (SYSOUT=* or DSN=) or the remote console session device.

SYSCONS - Console messages are written to SYSCONS based upon the SHOWCONS, SHOWON|OFF and SHOWERR start parameter options as well as the ALRMFLTR= filtering option. These options/parameters are described below:

SHOWCONS parameter or SHOW CONS ON command	controls the routing of all solicited (console command display output) to SYSCONS.
SHOWON|OFF parameter or SHOW ON|OFF command	permits the routing of all unsolicited event alert **informational** or **error** messages to SYSCONS.
SHOWERR parameter or SHOW ERR command	permits the routing of all unsolicited event alert **error** messages to SYSCONS except for special Informational Event/Alert Message Exceptions (NASnnnnI), see section below.
-	-
ALRMFLTR=	**Allow** (route to SYSCONS and SYSPRINT), **Suppress** (route to SYSPRINT only) or **Purge** (delete from SYSCONS or SYSPRINT) option controls the message disposition.

SYSPRINT - All solicited or unsolicited console messages are written to SYSPRINT although Informational NASnnnnI message can be filtered when the following options/parameters are employed:

PRNTON|OFF parameter or PRNT ON|OFF command	This command is used to start (ON) or stop (OFF) **all** SYSPRINT message logging (NAS*cnnns*) and console display output logging. PRNT OFF should only be used when alarm message and console display output is written to SYSCONS. If PRNT OFF and SHOW OFF are in effect, HNAS cannot report errors or console display output. It can, however, accept commands like PRNT ON.
PRNT*type* parameter or PRNT *type* ON|OFF command *type* = LU, QLLC, TCP, VC, VTAM, XOT and XTP.	The PRNT parameters control the routing of Informational messages (**NAS*cnnn*I**) to SYSPRINT based upon the message ***c***ategory type (i.e. NAS2nnnI, NAS3nnnI, etc.). Some special NAS*cnnn*I Informational Event/Alert Message are routed to SYSPRINT regardless of the PRNT setting (see Informational Event/Alert Message Exception section below for additional information).
-	-
ALRMFLTR=	Same rules as described in the SYSCONS table above.

Default Alarm Message Output Destination

Starting with the V2R2M0 release of HNAS, informational alarm message output is *not* routed to SYSCONS unless the SHOWON option is in effect. Only error alarm messages will be written to SYSCONS because the SHOWERR option is set by default. However, alarm messages are always written to the SYSPRINT log file regardless of the SHOW setting if the PRNT option is in effect. To allow all alarm messages be written to both SYSPRINT and SYSCONS, specify SHOWON as an HNAS start parameter or issue the SHOW ON console command. For additional information, please refer to the description of the HNAS start parameters in Chapter 2 of the HNAS Guide and Reference manual and the SHOW and PRNT console commands in this document.

The alarm filter parameter ALRMFLTR=(PURGE|SUPPRESS|ALLOW) also affects alarm output activity and routing. Please refer to the description of the alarm filter parameter ALRMFLTR= under Chapter 4 section BUILD of the HNAS Guide and Reference manual.

Informational Event/Alert Message Exceptions (NAS*cnnn*I)

Some HNAS (NAS*cnnn***I**) **I**nformational event and alert messages are forced to SYSCONS and SYSPRINT regardless of the SHOWERR (SYSCONS) or PRNTtype (SYSPRINT) message filtering mode.

This is because some Informational messages report successful completion of operational tasks or events and are considered to be important informational messages for displaying on the system operator console (SYSCONS).

A complete list of these special messages can be found in the **Informational Alert Message Considerations** section, Alert Messages Chapter of the HNAS Messages and Codes Guide.

Default Console Command Output Destination

Starting with the V2R2M0 release of HNAS, console command output is *not* routed to SYSCONS unless the SHOWCONS option is in effect. However, console command output is always written to the SYSPRINT log file regardless of the SHOW setting if the PRNT option is in effect. To allow console command output to be written to both SYSPRINT and SYSCONS, specify SHOWCONS as an HNAS start parameter or issue the SHOW CONS ON console command. For additional information, please refer to the HNAS start parameter descriptions in Chapter 2 of the HNAS Guide and Reference manual and SHOW and PRNT console command descriptions in this document.

Default Console Display Command Output

HNAS display console commands '**DMAP APAR**' and **DNAS** are now executed during HNAS activation regardless of the CONCMDQ= settings to ensure that support information is always available in the SYSPRINT output. APAR 2300192 was implemented to ensure that the DNAS command always executes and to disable the DMAP APAR output.

Netview Console Command Output Routing

HNAS console command output is solicited (uses synchronous WTOs) and as such is considered to be a response to a request. Console command output WTOs are routed to the requesting console using the CONSID=, CONSNAME= and/or CART= WTO operands. The requesting console can be the systems console (SYSCONS), a TSO user with console authority or a Netview console. The CONSID=, CONSNAME= or CART= values come from the request itself so the response is always guaranteed to get back to the requesting console. This routing is automatic because standard operating system interfaces are used. No additional system configuration is required, however, the HNAS SHOWCONS parameter must be in effect. This parameter can be specified as an EXEC start parameter (PARM='...,SHOWCONS,...') or via the SHOW CONS ON console command.

Netview Alarm Message Output Routing

HNAS alarm messages are unsolicited (use asynchronous WTOs) and thus require some system configuration changes to get them routed to Netview. All HNAS alarm messages start with the 3 character 'NAS' identifier (for example, NAS2021W). There are a few ways to route unsolicited WTOs to Netview:

1) Issue the Netview ASSIGN command as follows:

```
ASSIGN MSG=NAS*,PRI=OPER1
ASSIGN MSG=NAS*,COPY=OPER1
```

Any message that starts with 'NAS' should be routed to OPER1 if OPER1 is defined as a Netview console in the DSIOPF member of the NETVIEW.DSIPARM library and is an active console.

2) Add the following clause to the Netview automation table.

```
IF MSGID= 'NAS' . THEN
BEGIN;
ALWAYS
DISPLAY(Y) NETLOG(Y) SYSLOG(N)
END;
```

This change will route alarm messages to the Netview NETLOG but withhold them from the system log (SYSLOG). However, if SYSLOG is defined as a system HARDCOPY device, alarm messages will be routed to it before they can be filtered by the Netview automation table. To ensure that alarm messages only go to NETLOG, you will need to provide HNAS with a WTO routing code that is not subject to HARDCOPY routing. This is accomplished by specifying **OPTIONS=WTOROUTCDE(ALRM)=*value*** (we recommend *value*=11 => programmer information) on the BUILD definition statement and coding the following in the **CONSOL*xx*** member in the SYS1.PARMLIB library for the

SYSLOG HARDCOPY console:

ROUTCODE(1-10,12-128)

This will route all WTOs except ROUTCODE=11 to the HARDCOPY console but will require a system IPL to activate. To accomplish the same thing immediately, the following system command can be used:

VARY SYSLOG,HARDCPY,DROUT=(11)

This drops ROUTCDE=11 from the HARDCOPY ROUTCODE list but will only last until the system is re-IPLed. So the permanent change to CONSOL*xx* is required.

Note: All HNAS WTOs use a default routing code of 8 (teleprocessing control). This routing code along with 1,2,3,4,7,10 and 42 cannot be dropped from the HARDCOPY ROUTCODE= list. For this reason we recommend specifying a WTOROUTCDE= value that is unused and not one of those listed above.

3) To route error alarm messages, the HNAS SHOWERR parameter must be in effect. This parameter can be specified as an EXEC start parameter (PARM='...,SHOWERR,...') or via the SHOW ERR console command. Note that SHOWERR is a default HNAS start parameter so that technically it does not have to be specified in the PARM= operand.

Note: There are some exceptions for **'I' messages** that can be found in the HNAS Messages and Codes Guide Alert Messages sections 'Informational Alert Message Considerations'. It currently indicates that the messages: NAS0001I, NAS0910I, NAS3798I and NAS3799I will be sent to SYSCONS (and by extension NETVIEW) even though SHOWERR is in effect. These are considered important informational messages that must be shown to the systems operator. For additional information and a complete list of these special message exceptions please refer to the **Informational Alert Message Considerations** section, Alert Messages Chapter of the HNAS Messages and Codes Guide.

4) To route all alarm messages (error and informational alarms), the HNAS SHOWON parameter must be in effect. This parameter can be specified as an EXEC start parameter (PARM='...,SHOWON,...') or via the SHOW ON console command.

5) The HNAS PFXWTO parameter can be used to prefix all HNAS alarm messages with either the BUILD NASNAME= operand value (specify PFXWTO with no follower) or with any text string (specify PFXWTO followed by a *text* string). For example, specify EXEC PARM='...,PFXWTO *text*,... or issue the PFXWTO *text* console command. If the PFXWTO parameter is used, the MSG= operand for the ASSIGN command or the MSGID= operand for the IF/THEN clause above will have to be changed to specify the WTO prefix text (either the NASNAME= operand value or *text*).

Local console name in alarm messages - WACONPCE

WACONPCE is the name of the local console PCE. If a command is entered that changes the state of a HNAS component from the SYSCONS or by a TSO user with console authority, WACONPCE will appear in an alarm message so the operations personnel can determine who issued the command. For example, if **TRCALL SUSP** was entered from the SYSCONS, the following message is displayed:

```
NAS0050A TRACING SUSPENDED BY WACONPCE
```

Remote console name in alarm messages - RCON*nnnn*

RCON*nnnn* is the name of a remote console PCE (*nnnn*= 0001 to the CONLMT= value). If a command is entered that changes the state of a HNAS component from a remote console, RCON*nnnn* will appear in an alarm message so the operations personnel can determine who issued the command. For example, if **TRCALL RSME** was entered from the first allocated remote console, the following message is displayed:

```
NAS0060W TRACING RESUMED BY RCON0001
```

Table Display

Many display commands produce tabular output in row/column format. For some of these commands, the type of data that is displayed can be modified using command parameters. The following commands produce tabular output: DLU, DMCH, DPCE, DSTATS and DVC.

Echoing Command Input

When a console operator enters a command, HNAS echoes the command back to console (preceded by the text **HNASCMD->**) so that it is recorded in the log for the console session. For example, if CONPRMT=HNAS240-> and the console operator enters TRCMCH ALLON, the following will be written to the log:

```
HNAS240-> TRCMCH ALLON                    <- operator's input after prompt
HNASCMD-> TRCMCH ALLON                    <- input echoed back by HNAS
```

Echoing Queued Command Input

The **HNASCMD->** command prefix can now be replaced with EXECCMD->, SCHDCMD->, PRXY-CMD-> or TRAPCMD-> depending on how the command was scheduled for execution.

`EXECCMD-> =>` command came from the CONCMDQ= operand or EXEC command.
`SCHDCMD-> =>` command came from the SCHEDULE= operand or SCHEDULE command.
`PRXYCMD-> =>` command came from another console.
`TRAPCMD-> =>` command came from a trapped event.

This change requires the use of bits 0 and 1 in the command string length byte for all queued commands which restricts each queued command string length to 63-bytes.

Note: HNASCMD-> overlay support was introduced into 240 with Enhancement APAR 2400075.

Echoing Command Execution Parameters

HNAS will echo back to the console operator (preceded by the text HNASXEQ->) the command modifiers and parameters (default and specified) to show the operator what values are being used to execute the command. This also provides an audit log entry for the console session. For example, if the console operator enters TRCMCH MCH1 ON LCL ICR OCR, the following will be written to the log:

```
HNAS240-> TRCMCH MCH1 ON LCL ICR OCR             <- operator's input after prompt
HNASCMD-> TRCMCH MCH1 ON LCL ICR OCR             <- input echoed back by HNAS
HNASXEQ-> TRCMCH RNM=MCH1 ON LCL ICR ON OCR ON   <- command as processed
```

Blank Column in Tabular Display

When a column field in a display table is not applicable to a component, the field will display as blank. For example, if DLU SHOWNAME=RTR is specified and no router connection is active for an SLU, the RTRNAME/ADDRESS column in the DLU display will be blank for the SLU. Another example, if DVC SHOWLUST is specified and no SLU connection is active for the VC, the LUST column in the DVC display will be blank for the VC.

Numeric Display

Most numeric output for all display commands is in **hexadecimal** unless noted otherwise. Please keep this in mind when comparing HNAS console display and trace output with network or Cisco router debug event logs which are typically in decimal. For example: HNAS PVC LCN's are represented in hexadecimal while Cisco router PVC LCN's show or debug displays are represented in decimal).

Aborting Output

Any form of input can be used to terminate output. For example, a null input line or BREAK from a remote console will result in stopping output and forcing the input prompting message (e.g., **ENTER COMMAND:**) using the text supplied by the **CONPRMT** operand on the BUILD definition statement. When input aborts a running console command, the following message is issued:

```
NASC005W command COMMAND ABORTED DUE TO OPERATOR INPUT
```

If the aborted command happens to be an element of a command list that is being executed, the following additional message will be issued:

```
NASC006W cmdlist COMMAND LIST INTERRUPTED
```

NEWDEFN Output

A new definition (NEWDEFN) CDF can be created from the original CDF with changes incorporated using the MLCL and/or MRMT command. The **NEWDEFN DD** statement points at the new CDF image (it should not point at the original CDF unless you want to replace it). If no changes are made using MLCL and/or MRMT, the new CDF will be identical to the original except for comments immediately preceding each definition statement. For example:

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123
*BUILD=nasname                                                        ;NWDF
*LOCAL=lclname                                                        ;NWDF
*REMOTE=rmtname                                                       ;NWDF
```

If changes are made using MLCL and/or MRMT, the changed or added statements will have the **;NWDF** eyecatcher starting in record column 67. HNAS attempts to preserve as much of the existing comment as possible (starting with the first semicolon ;) when a record is modified.

Note: The NEWDEFN file is maintained in memory until HNAS is stopped at which time it is then written to disk. You can display the updated NEWDEFN file in memory by issuing the DNWDF command which is described on page CONS-89.

Note: NEWDEFN support is provided only when the **GENNWDF** start parameter is specified and the NEWDEFN DD statement is included in the HNAS start JOB.

Command Description Rules

CAPITAL LETTERS

represent values that you enter as given, without change.

small italicized letters

represent parameters for which you must supply a value.

brackets [ ]

enclose operands or symbols that are either optional or conditional.

An optional parameter is one that you can code or omit, without regard to the coding of other operands.

A conditional parameter is one that you can code or omit, depending how you code or omit other operands for the same command.

braces { }

indicate that you must choose a value from a list of values enclosed within the braces.

single vertical bar | (OR symbol)

indicates that you must choose a value from the values on either side of the OR symbol.

double vertical bars || (concatenation symbol)

indicate that data separated by the concatenation symbol will be appended to one another.

ellipsis ... (3 consecutive dots)

indicates that you can code the preceding item more than once. For example, PADPARM=(*pn*1,*pv*1,...,*pn*i,*pv*i)).

Miscellaneous Rules

Console Command versus Start Parameter

Many console commands provide the same functions as HNAS start parameters. For a cross reference list of start parameters and their console command equivalents, please see Table 3 in Chapter 2.

Console Command Parameters versus Command Actions

Most console commands perform actions as a function of the command parameters provided but the actions can be restricted based on the type of resource being acted upon. An **action matrix** is provided for a number of these commands that show the command parameter, command action and the resource type for which it applies. For examples, see the DLCL, DRMT, MLCL, MRMT and SHOW commands.

Console Command Parameter Action Matrix Keys

blank	indicates that the command parameter is **valid** for the resource type
I	indicates that the command parameter is **invalid** for the resource type
N	indicates that the command parameter is **ignored** for the resource type
S	indicates that the command action is **suppressed**

Console Command Change Control/Version, Release, Modification

CC/VRM heading entry values are used to denote **C**hange **C**ontrol activity for the HNAS **V**ersion, **R**elease and **M**odification level where the changes were introduced. The CC/VRM value identifiers are typically assigned for changes affecting operand's, parameter's or feature's under the following circumstances: A **blank or (N)** CC value denotes that the enhancement was introduced in the initial release of the denoted VRM distribution level. A value of **(A) Added**, **(C) Changed**, **(D) Deleted** or **(R) Retired** denotes change control activity after the initial release.

For CR/VRM headings that are listed against operands with multiple suboperands, please refer to the specific operand table in Chapter 4 of the HNAS Guide and Reference (for the BUILD, LOCAL or REMOTE definition statement) where the parameter is described. This will allow you to identify CC/VRM changes at the suboperand level.

How To Index Entries

Please refer to the index at the end of this document for references to sections which provide examples of ‘how to’ use console commands to effect dynamic changes or display HNAS information.

Command Overview

This section of the Console Subsystem Operations Guide provides an overview of the Display, Modify, Trace and Statistics Commands that are recognized by the HNAS Console Subsystem

Note: Privileged commands listed in this section are underlined.

Display Commands

Display Commands	Description	cc/vrm
ALARM	Display alarm information	**C/220**
D or DMEM	Display memory	
DADDR	Display TRCADDR address parameters	
DLCL	Display LOCAL configuration	**N/220**
DLP	Display HNAS load point	
DLU	Display Logical Unit	**C/220** **C/240**
DMAP	Display HNAS module map	**C/220**
DMCH	Display Multi CHannel Link	**C/240**
DNAS	Display HNAS information	**C/220** **C/230** **C/240**
DNWDF	Display NEWDEFN CDF	**A/230**
DPARM	Display command modifiers	**C/220**
DPCE	Display Process Control Element (PCE)	**C/220** **C/240**
DRMT	Display REMOTE configuration	**C/220** **C/230**
DSTATS	Display PCE statistics	
DTRC	Display trace table	
DVC	Display Virtual Circuit	**C/220**
F or FIND	Find data in memory	
? or HELP	Display command HELP	**C/211** **C/230**

Console Subsubsystem

Display Commands	Description	cc/vrm
MON	Display HNAS utilization or REMOTE Keep Alive processing	**C/210** **C/230**
PRNT	Display active SYSPRINT log (DDNAME=?) or PRTSWLST= list (PRTSWLST=?)	**C/240**
TRCTRAP	Display trace trap information	**A/230**
WRU	Display NASNAME	**N/230**

Control and Modify Commands

Control and Modify Commands	Description	cc/vrm
ALARM	Alarm control	**C/220**
ECHOXEQ	HNASXEQ echo data control	**A/240**
END	Effective NOP. Normally used to delimit an EXEC command list	**A/240**
EXEC	Execute a list of commands	**N/240**
M or MMEM	Modify memory	**C/240**
MLCL	Modify LOCAL configuration	**N/220** **C/240**
MON	Monitor HNAS utilization or REMOTE Keep Alive processing	**C/210** **C/230**
MRMT	Modify REMOTE configuration	**C/220** **C/230** **C/240**
PAUSE	Provide forced delay	**N/240**
PFXWTO	WTO prefix control	**C/230**
PING	Ping remote router	**N/230** **C/240**
PRNT	SYSPRINT control	**C/220**
Q or QUIT	Terminate operations	**C/220** **C/240**
RMTCONS	Remote console control	
SCHEDULE	Schedule commands for execution at specific times	**A/240**
SHOW	Console display control	**C/220**
SMSG	Send message to other console	
SNAP	Dump HNAS memory areas to SYSPRINT	**A/240**
STATS	PCE statistics control	
TRCPRNT	Trace SYSPRINT control	**C/220**
TRCTRAP	Trace trap control	**A/220** **C/240**

Control and Modify Commands	Description	cc/vrm
V or VARY	Vary resource state	**C/220** **C/240**

Trace Commands

Trace Commands	Description	cc/vrm
TRCADDR	Resource trace control	
TRCALL	All trace control	**C/230** **C/240**
TRCBFR	PCE buffer trace control	**C/230** **C/240**
TRCBFRQ	Buffer request trace control	
TRCBST	Binary search trace control	**N/220**
TRCCNFG	Configuration trace control	**N/230**
TRCCONS	Console event trace control	**A/230**
TRCDATA	PCE data trace control	**C/230** **C/240**
TRCDBK	LU/VC data block trace control	**N/230**
TRCDISP	PCE dispatcher trace control	**C/230** **C/240**
TRCIO	PCE input/output trace control	**C/230** **C/240**
TRCLU	LU trace control	**C/220**
TRCLUQ	LU allocation queue trace control	**C/240**
TRCMCH	MCH trace control	**C/240**
TRCMCHX	MCHX trace control	**C/220** **C/240**
TRCPCE	PCE trace control	**A/230**
TRCPRNT	SYSPRINT trace control	**C/220**
TRCSUBR	Subroutine trace control	**C/240**
TRCTASK	Task WAIT/POST trace control	**C/210**
TRCTRAP	Trace trap control	**A/230**
TRCVC	VC trace control	**C/220**
TRCVCQ	VC allocation queue trace control	**C/240**
TRCWTO	WTO request trace control	

Statistics Commands

Statistics Commands	Description	cc/vrm
DSTATS	Display PCE statistics	
MON	Display HNAS utilization	**C/210** **C/230**
STATS	PCE statistics control	**C/240**

Command Modifier Overview

This section of the Console Subsystem Operations Guide provides an overview of the Command Modifiers that are recognized by the HNAS Console Subsystem

Console command modifiers provide command arguments that can be permanently set (like a REMOTE name via the RNM= modifier) when they are entered by themselves (stand alone) or on the left side of a command. The permanent modifies are copied into a temporary workarea prior to execution of each command. Commands only use temporary modifiers. Temporary modifiers can be overridden if a modifier is entered on the right side of a command. For example:

```
RNM=MCH1              <- Set permanent REMOTE name remembrance
DRMT                  <- Display configuration information for MCH1
DRMT RNM=MCH2         <- Display configuration information for MCH2
DLU                   <- Display all LUs on MCH1
```

The temporary modifiers can be reset (set to their null values) if the **BPM** parameter is specified immediately after a command (right side modifier only). This can be used to change the way the command is executed. For example:

```
RNM=MCH1              <- Set permanent REMOTE name remembrance
DRMT                  <- Display configuration information for MCH1
DRMT BPM              <- Displays the names of all REMOTEs when RNM= is not set
```

Note: The BPM parameter was introduced with enhancement APAR 2400014.

Note: When a REMOTE name, LOCAL name or SLU name is specified as a right side modifier, the modifier keyword (RNM=, LNM= or LUNM=, respectively) is not required to precede the name. This is because the modifier keyword is implied by the type of resource that is named. For example: `RNM=MCH2` is implied when `DRMT MCH2` is entered. In other words, `DRMT MCH2` and `DRMT RNM=MCH2` are treated identically.

Note: An error message will be generated if the wrong type of resource is identified for a given modifier. For example, if `LUNM=MCH1` is specified and MCH1 is actually a REMOTE name, the following error message is generated:

```
NASC410E LUNM=MCH1 INVALID, REQUIRED
```

Note: An error message will be generated if a command requires a resource and the resource has not been named by a command modifier. For example, if `RNM=MCH1` is specified as a permanent (stand alone or left side modifier) and `MRMT BPM MBITCHN=YES` is specified, the following error message is generated:

```
NASC300E RNM= OMITTED, REQUIRED
```

This error occurs because the BPM parameter resets the RNM=MCH1 modifier that was copied to the temporary workarea.

Command Modifier Summary (Alphabetical Order)

Command Modifiers	Description	cc/vrm
BPM	Resets all temporary modifiers for the duration of the current command only (must be the first parameter specified after the command)	**A/240**
CID=	Sets LUB identifier(s) for DLU, TRCLU	
CLDADDR=	Sets *Called* DTE address for TRCADDR, PING	
CLGADDR=	Sets *Calling* DTE address for TRCADDR, PING	
ID=	Sets PCE identifier(s) for DLU, DMCH, DPCE, DSTATS, DVC, MON, SMSG, STATS, TRCBFR, TRCCONS, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ, VARY	
IFN=	Sets MCH identifier(s) for DLU, DMCH, DVC, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCVC, TRCVCQ **Note:** IFN= is used for XTP only.	
IPADDR=	Sets IP address for TRCADDR, PING	
LNCT=	Sets remote console (RMTCONS) scroll line count	
LNM=	Sets LOCAL name for DLCL, DNWDF, MLCL, VARY	**N/220**
LUN=	Sets MCHX (CTCP SLU) identifier(s) for DMCH, TRCLUQ, TRCMCHX	
LUNM=	Sets LU name for DLU, TRCLU, TRCMCHX, VARY	**N/220**
RNM= or RNAME=	Sets REMOTE name for DLU, DMCH, DNWDF, DPCE, DRMT, DSTATS, DVC, MON, MRMT, PING, STATS, TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ, VARY	**C/220**
VCN=	Sets VCB identifier(s) for DVC, TRCVC	

Command Modifier Hierarchy Matrix By Command

Precedence Rank	Command Modifiers	Command Usage
1	**LUNM=**	Overrides RNM=, CID=, LUN=, IFN= and ID= for DLU, TRCLU, TRCMCHX, VARY LU
2	**RNM= or RNAME=**	Overrides IFN= and ID= for DLU, DMCH, DPCE, DRMT, DSTATS, DVC, MRMT, PING, STATS, TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ, VARY RMT
2	**LNM=**	Overrides ID= for VARY LCL
3	**IFN=**	Used only when RNM= is not set for DLU, DMCH, DVC, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCVC, TRCVCQ **Note:** IFN= is used for XTP only.
3	**LUN=**	Used only when LUNM= is not set for TRCMCHX
3	**CID=**	Used only when LUNM= is not set for DLU, TRCLU
3	**VCN=**	Used always regardless of RNM= or ID= setting for DVC, TRCVC
4	**ID=**	Used only when RNM= is not set for DLU, DMCH, DPCE, DSTATS, DVC, MON, SMSG, STATS, TRCBFR, TRCCONS, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ, VARY RMT or when LNM= is not set for VARY LCL

Note: The CLDADDR=, CLGADDR=, IPADDR= and LNCT= modifiers are independent from those listed in the above table.

Command Modifier Hierarchy Matrix By Resource Type

Precedence Rank	All but LOCAL resources	For LOCAL Resources
1	LUNM=	
2	RNM= or RNAME=	LNM=
3	IFN=, LUN=, CID=, VCN=	
4	ID=	ID=

Command Modifier/Control Block Relationship

The following sections describes the command modifiers as they relate to HNAS resource control blocks. To access a HNAS resource for display, modification and trace processing, you must use a command modifier that represents the resource's control block.

Process Control Element Identification (ID=)

Process Control Elements (PCEs) are the control blocks that are used to manage HNAS sub-tasks and represent both LOCAL and REMOTE resources that require WAIT/POST event processing. Commands relating to HNAS PCEs use the **ID=*minid-maxid*** modifier to identify specific PCEs. The ID= value you enter remains in effect until a new value is entered or until your console session is terminated.

Note: The ID= modifier is used by the DLU, DMCH, DPCE, DSTATS, DVC, MON, SMSG, STATS, TRCBFR, TRCCONS, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ and VARY commands.

If ID= (null) is specified, the command operates on all PCEs unless noted otherwise.

If ID=0 is specified (ID=0-0 is not permitted), the command also operates on all PCEs unless noted otherwise. For most commands, ID= (null) and ID=0 are treated identically. The exceptions are for the TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ, VARY and MON TAP commands. Please refer to description of these commands for further details.

If an ID= value is specified as a range (e.g., ID=1-3), the command operates on the selected PCEs.

If an ID= value is specified as a single element (e.g., ID=1) or as a range where the extents are the same (e.g., ID=1-1), the command operates on the selected single PCE.

If an invalid ID= value is specified, the balance of the command line is ignored.

If an ID= value is specified, it must be entered as a hexadecimal number.

ID=*minid*[-*maxid*] extent values can range from 1 to *n* where *n* is the last configured PCE. To see the value of *n*, issue the DPARM command and look under the VALID VALUES column for the ID= modifier.

A PCE is the control block used to manage HNAS subtasks. The following table lists the HNAS PCEs and the internal ID numbers assigned to them. For the ID values in the table below:

xtpl=#XTP_LOCAL_resources, *xtpr*=#XTP_REMOTE_resources
xotl=#XOT_LOCAL_resources, *xotr*=#XOT_REMOTE_resources (sum of XOT VCLMT= values)
mchr=#MCH_REMOTE_resources, *spur*=#SPU_REMOTE_resources
dflr=#DFL_REMOTE_resources, *dfsr*=#DFS_REMOTE_resources

Process Control Element Identification (ID=) Table

Type	ID=
HNAS TIMER	1
HNAS LOCAL CONSOLE	2
HNAS NETVIEW	3 (reserved)
HNAS PING	4
HNAS UTILITY	5
XTP UTILITY	6 if *xtpl*>0
XTP LOCAL TCP	7 if *xotl*=0 or 8 if *xotl*>0 for *xtpl* PCEs
XTP REMOTE TCP	7+*xtpl* if *xotl*=0 or 8+*xtpl*+*xotl* if *xotl*>0 for *xtpr* PCEs
XOT UTILITY	6 if *xtpl*=0 or 7 if *xtpl*>0
XOT LOCAL TCP	7 if *xtpl*=0 or 8+*xtpl* if *xtpl*>0 for *xotl* PCEs
XOT REMOTE TCP	7+*xotl* if *xtpl*=0 or 8+*xtpl*+*xtpr*+*xotl* if *xtpl*>0 for *xotr* PCEs
MCH REMOTE	7+*xotl*+*xotr* if *xtpl*=0 or 8+*xtpl*+*xtpr*+*xotl*+*xotr* if *xtpl*>0 for *mchr* PCEs
SPU REMOTE	7+*xotl*+*xotr*+*mchr* if *xtpl*=0 or 8+*xtpl*+*xtpr*+*xotl*+*xotr*+*mchr* if *xtpl*>0 for *spur* PCEs
DFL REMOTE	7+*xotl*+*xotr*+*mchr*+*spur* if *xtpl*=0 or 8+*xtpl*+*xtpr*+*xotl*+*xotr*+*mchr*+*spur* if *xtpl*>0 for *dflr* PCEs
DFS REMOTE	7+*xotl*+*xotr*+*mchr*+*spurr*+*dflr* if *xtpl*=0 or 8+*xtpl*+*xtpr*+*xotl*+*xotr*+*mchr*+*spur*+*dflr* if *xtpl*>0 for *dfsr* PCEs
REMOTE CONSOLE	7+*xotl*+*xotr*+*mchr*+*spurr*+*dflr*+*dfsr* if *xtpl*=0 or 8+*xtpl*+*xtpr*+*xotl*+*xotr*+*mchr*+*spur*+*dflr*+*dfsr* if *xtpl*>0 for CONLMT PCEs

Note: The **DPCE BPM** command may be used to display all PCE identifiers, names and state information.

XTP Multi-channel Link Block Identification (IFN=)

XTP Multi-channel Link Blocks (MCHs) are the control blocks that are used to manage physical XTP MCH activity. Commands relating to HNAS XTP MCHs use the **ID=*minid-maxid*** modifier to identify specific PCEs and the **IFN=*minifn-maxifn*** modifier to identify specific MCHs. The ID= and IFN= values you enter remain in effect until new values are entered or until your console session is terminated.

Note: The IFN= modifier is used by the DLU, DMCH, DVC, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCVC and TRCVCQ commands.

If ID= (null) or ID=0 is specified, the command operates on all PCEs unless noted otherwise.

If IFN= (null) is specified, the command operates on all MCHs unless noted otherwise.

If IFN=0 is specified (IFN=0-0 is not permitted), the command also operates on all MCHs unless noted otherwise. For all commands, IFN= (null) and IFN=0 are treated identically.

If an IFN= value is specified as a range (e.g., IFN=2-17), the command operates on the selected MCHs.

If an IFN= value is specified as a single element (e.g., IFN=2) or as a range where the extents are the same (e.g., IFN=2-2), the command operates on the selected single MCH.

If an invalid IFN= value is specified, the balance of the command line is ignored.

If an IFN= value is specified, it must be entered as a decimal number.

IFN=*minifn*[-*maxifn*] values can range from 1 to 255.

Example: To view the status of the MCH interface 1 on the first TYPE=XTP REMOTE router, enter '**`ID=6 IFN=1 DMCH`**'.

Note: The IFN= modifier is used only when the ID= modifier identifies a TYPE=XTP REMOTE definition statement. HNAS generates a simulated NPSI Multi-channel Link Block (MCH) for each TYPE=XTP REMOTE definition statement. HNAS also generates an MCH for each TYPE=MCH REMOTE definition statement. The latter, however, is referenced directly by the ID= modifier (the IFN= value does not apply).

The MCH is the control block used to manage HNAS Multi-channel Link activity. The IFN= value is the interface number that identifies a physical X.25 link on a remote XTP router.

MCH Control Session Logical Unit Block Identification (LUN=)

Multi-channel Link Extension Blocks (MCHXs) are the control blocks that are used to manage MCH CTCP Control Session SLU activity. Commands relating to HNAS MCH Control Session Logical Unit Blocks (Control LUBs) or MCH Extension Blocks (MCHXs) as they are also called, use the **ID=*minid-maxid*** modifier to identify specific PCEs, the **IFN=*minifn-maxifn*** modifier to identify specific MCHs (XTP only) and the **LUN=*minlun-maxlun*** modifier to iden-

tify specific MCHXs. The ID=, IFN= and LUN= values you enter remain in effect until new values are entered or until your console session is terminated.

Note: The LUN= modifier is used by the DMCH, TRCLUQ and TRCMCHX commands.

If ID= (null) or ID=0 is specified, the command operates on all PCEs unless noted otherwise.

If IFN= (null) or IFN=0 is specified, the command operates on all MCHs unless noted otherwise.

If LUN= (null) is specified, the command operates on all MCHXs unless noted otherwise.

If LUN=0 is specified (LUN=0-0 is not permitted), the command also operates on all MCHXs unless noted otherwise. For all commands, LUN= (null) and LUN=0 are treated identically.

If a LUN= value is specified as a range (e.g., LUN=1-3), the command operates on the selected MCHXs.

If a LUN= value is specified as a single element (e.g., LUN=1) or as a range where the extents are the same (e.g., LUN=1-1), the command operates on the selected single MCHX.

If an invalid LUN= value is specified, the balance of the command line is ignored.

If a LUN= value is specified, it must be entered as a decimal number.

LUN=*minlun*[-*maxlun*] values can range from 1 to 28.

Example: To view the status of the first CTCP SLU for MCH interface 1 on the first TYPE=XTP REMOTE MCH, enter '**`ID=6 IFN=1 LUN=1 DMCH`**'.

Note: HNAS generates an MCH Extension Block (MCHX) for each SLU name in the LUNAME= operand list for a TYPE=XTP|MCH REMOTE definition statement. The MCHX is the control block used to manage HNAS CTCP control logical unit sessions. The LUN= value identifies a specific CTCP SLU listed in the LUNAME operand for an MCH (1 through 28).

MCH Terminal Session Logical Unit Block Identification (CID=)

Logical Unit Blocks (LUBs) are the control blocks that are used to manage SLU activity. Commands relating to HNAS MCH Terminal Session LUBs (Terminal LUBs) use the **ID=*minid-maxid*** modifier to identify specific PCEs, the **IFN=*minifn-maxifn*** modifier to identify specific MCHs (XTP only) and the **CID=*mincid-maxcid*** modifier to identify specific LUBs. The ID=, IFN= and CID= values you enter remain in effect until new values are entered or until your console session is terminated.

Note: The CID= modifier is used by the DLU and TRCLU commands.

If ID= (null) or ID=0 is specified, the command operates on all PCEs unless noted otherwise.

If IFN= (null) or IFN=0 is specified, the command operates on all MCHs unless noted otherwise.

If CID= (null) is specified, the command operates on all LUBs unless noted otherwise.

If CID=0 is specified (CID=0-0 is not permitted), the command also operates on all LUBs unless noted otherwise. For all commands, CID= (null) and CID=0 are treated identically.

If a CID= value is specified as a range (e.g., CID=3-10), the command operates on the selected LUBs.

If a CID= value is specified as a single element (e.g., CID=3) or as a range where the extents are the same (e.g., CID=3-3), the command operates on the selected single LUB.

If an invalid CID= value is specified, the balance of the command line is ignored.

If a CID= value is specified, it must be entered as a hexadecimal number.

CID=*mincid*[-*maxcid*] values can range from 1 to FFFFFFFE.

Example: To view the status of the first 10 data logical units for MCH interface 1 on the first TYPE=XTP REMOTE router, enter '**`ID=6 IFN=1 CID=1-10 DLU`**'.

Note: HNAS generates an Logical Unit Block (LUB) for every SLU that can be supported across all router connections. A LUB is the control block used to manage HNAS terminal logical unit sessions. The CID= value is a number assigned by VTAM as logical units activate.

MCH Virtual Circuit Block Identification (VCN=)

Virtual Circuit Blocks (VCBs) are the control blocks that are used to manage VC activity. Commands relating to HNAS MCH VCBs use the **ID=*minid-maxid*** modifier to identify specific PCEs, the **IFN=*minifn-maxifn*** modifier to identify specific MCHs (XTP only) and the **VCN=*minvcn-maxvcn*** modifier to identify specific VCBs. The ID=, IFN= and VCN= values you enter remain in effect until new values are entered or until your console session is terminated.

Note: The VCN= modifier is used by the DVC and TRCVC commands.

If ID= (null) or ID=0 is specified, the command operates on all PCEs unless noted otherwise.

If IFN= (null) or IFN=0 is specified, the command operates on all MCHs unless noted otherwise.

If VCN= (null) is specified, the command operates on all VCBs unless noted otherwise.

If VCN=0 is specified (VCN=0-0 is not permitted), the command also operates on all VCBs unless noted otherwise. For all commands, VCN= (null) and VCN=0 are treated identically.

If a VCN= value is specified as a range (e.g., VCN=3-10), the command operates on the selected VCBs.

If a VCN= value is specified as a single element (e.g., VCN=3) or as a range where the extents are the same (e.g., VCN=3-3), the command operates on the selected single VCB.

If an invalid VCN= value is specified, the balance of the command line is ignored.

If a VCN= value is specified, it must be entered as a decimal number.

VCN=*minvcn*[-*maxvcn*] values can range from 1 to 4095.

Example: To view the status of the first 10 virtual circuits for MCH interface 1 on the first TYPE=XTP REMOTE router, enter '**`ID=6 IFN=1 VCN=1-10 DVC`**'.

Note: HNAS generates an Virtual Circuit Block (VCB) for every virtual circuit that can be supported across all router connections. The VCB is the control block used to manage HNAS virtual circuit sessions. The VCN= value is an internal number assigned sequentially by HNAS as virtual circuits activate.

Logical Unit Name (LUNM=)

(new for V2R2M0)

You can target a specific logical unit resource and override the current settings for ID=, IFN= and CID= or LUN= by specifying the LU name in the LUNM= modifier. LUNM= can reference any LU that is defined in the CDF. The LUNM= value you enter remains in effect until a new value is entered or until your console session is terminated. If the LUNM= modifier is entered with no value, the remembered LU name is reset.

Note: The LU resource named by the LUNM= modifier is used by the DLU, TRCLU, TRCMCHX and VARY LU commands.

Local Name (LNM=)

(new for V2R2M0)

You can target a specific local resource and override the current settings for ID= by specifying the LOCAL name in the LNM= modifier. LNM= can reference any LOCAL definition statement defined in the CDF. The LNM= value you enter remains in effect until a new value is entered or until your console session is terminated. If the LNM= modifier is entered with no value, the remembered LOCAL name is reset.

Note: The LOCAL resource named by the LNM= modifier is used by the DLCL, DNWDF, MLCL and VARY LCL commands.

Remote Name (RNM= or RNAME=)

(changed for V2R2M0)

You can target a specific remote resource and override the current settings for ID= and IFN= by specifying the REMOTE name in the RNM= modifier. RNM= can reference any REMOTE definition statement in the CDF. The RNM= value you enter remains in effect until a new value is entered or until your console session is terminated. If the RNM= modifier is entered with no value, the remembered REMOTE name is reset.

Note: The REMOTE resource named by the RNM= modifier is used by the DLU, DMCH, DNWDF, DPCE, DRMT, DSTATS, DVC, MON, MRMT, PING, STATS, TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ and VARY RMT commands.

Command and Command Modifier Descriptions

Command Header Description

Prefix Modifiers	Command	Suffix Modifiers followed by Command Parameters	CC	VRM
[*modifiers*]	*command*	[*modifiers*] [*parameters*]		*vrm*

Prefix Modifiers represent modifiers global to all related commands. Prefix modifiers remain in effect for the duration of the console session or until changed by the console operator.

Command represents the specific command required to achieve the desired results.

Suffix Modifiers override prefix modifiers for the duration of the current command only.

CC denotes the Change Control activity for the HNAS VRM level where changes were introduced or modified. The cc/vrm value identifiers are typically assigned for changes affecting commands, modifiers, formats or operational characteristics under the following circumstances: A blank or (N) CC value denotes that the enhancement was introduced in the initial release of the denoted vrm distribution level. A value of (A) Added, (C) Changed, (D) Deleted or (R) Retired denotes change control activity for an active vrm release.

VRM refers to the HNAS Version, Release, Modification level for the change control implementation type.

General notes on Command Headers:

1) Multiple commands can be entered on the same input line and are processed left to right. If a command list includes a display command, processing will stop after the display command completes. For example, TRCLU ON DLU TRCVC ON will cause the TRCLU and DLU commands to be executed but not the TRCVC command because DLU is a display command which terminates input parsing.

2) A prefix modifier can be entered **by itself** or on the **left side** of a command. Multiple prefix command modifiers can be entered on the same input line before (to the left of) the first command (or when no command is given). For example, ID=*id* CID=*cid* TRCLU ON.

3) A suffix modifier can only be entered on the **right side** of a command. Multiple suffix command modifiers can be entered on the same input line after (to the right of) a command but must be entered before any command arguments (parameters). For example, TRCLU ID=*id* CID=*cid* ON. If a suffix modifier follows a command argument (like ON in this example), it will be treated as a prefix modifier after the current command completes. The only exception to this rule is when the current command is a display command in which case, input parsing stops after the display command completes thus preventing the suffix (turned prefix) modifier from being processed.

ALARM Command - Alarm Selection Control (Privileged)

(changed for V2R1M0)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	ALARM	[{LCLCONS|*rconname*}]	**A**	**240**
		[{{START|STRT|ON}| {STOP|OFF}}] [SHOW]	**N** **A**	**210** **230**
		[FILTER={?| *| (A|S|P, *id*1(A|S|P|FC|FU), : *id*n(A|S|P|FC|FU))}]	**C** **A** **A** **A**	**220** **230** **240** **240**
		[LIMITS={?| *| (*interval* *ilmt*, *dlmt*, *almt*, *wlmt*, *elmt*. *slmt*, *mlmt*)}]	**N** **A**	**220** **230**
		[LOG={?| CLEAR|RESET}]	**N**	**220**
		[MSGTXT={?|SHORT|LONG}	**A**	**240**
		[PULSE={?| *| (*hh*:*mm*:*ss*, *hh*:*mm*:*ss*, *seconds*)]	**A**	**240**

This command is used to activate (**START|STRT|ON**) or deactivate (**STOP|OFF**) alarm monitor mode for a remote console session. For local or remote consoles, this command is also used to display (**SHOW**) all alarm parameters that are currently active, set, reset or display the alarms being monitored (**FILTER=(*id*1,...,*id*n)**, **FILTER=*** or **FILTER=?**), set, reset or display the alarm limits for all alarms (**LIMITS=(*interval*,*ilmt*,...,*mlmt*)**, **LIMITS=*** or **LIMITS=?**), clear, reset or display all entries in the alarm logging table (**LOG=CLEAR**, **LOG=RESET** or **LOG=?**), toggle or display the value for the BUILD OPTIONS=ALRMSGTXT= parameter (**MSGTXT=SHORT**, **MSGTXT=LONG** or **MSGTXT=?**) and/or set, reset or display HNAS

pulse (heartbeat) parameters (**PULSE=(*starttime*,*endtime*, *frequency*)**, **PULSE=*** or **PULSE=?**).

- When the **LCLCONS** or ***rconname*** argument is provided, it directs ALARM processing to the identified console. This allows one console to alter ALARM processing for another console. For example, ALARM RCON0031 STOP will terminate alarm mode for the remote console named RCON0031 regardless of which console actually entered the command. For more information on LCLCONS|*rconname* processing, please refer to section entitled 'Console PCE name as a Suffix Command Modifier' on page CONS-11.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

ALARM Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
START|STRT|ON	**STOP|OFF**	**LCLCONS|*rconname***	**1**
STOP|OFF	**START|STRT|ON**	**LCLCONS|*rconname***	**2**
FILTER=*	**none**	**LCLCONS|*rconname***	**3**
FILTER=*list*	**none**	**LCLCONS|*rconname***	**4**
LIMITS=*	**none**	**LCLCONS|*rconname***	**5**
LIMITS=*list*	**none**	**LCLCONS|*rconname***	**6**
MSGTXT=SHORT	**MSGTXT=LONG**	**system wide**	**7**
MSGTXT=LONG	**MSGTXT=SHORT**	**system wide**	**8**
LOG=RESET	**LOG=CLEAR**	**system wide**	**9**
LOG=CLEAR	**LOG=RESET**	**system wide**	**10**
PULSE=*	**none**	**system wide**	**11**
PULSE=*list*	**none**	**system wide**	**12**
SHOW or FILTER=?	**none**	**LCLCONS|*rconname***	**13**
SHOW or LIMITS=?	**none**	**LCLCONS|*rconname***	**14**
SHOW or MSGTXT=?	**none**	**system wide**	**15**
SHOW or LOG=?	**none**	**system wide**	**16**
SHOW or PULSE=?	**none**	**system wide**	**17**

Alarm Monitoring Commands

- Enter **ALARM START** to enable alarm monitoring for the issuing remote console

- Enter **ALARM STOP** to disable alarm monitoring for the issuing remote console.

- Enter **ALARM SHOW** to display the alarm parameters that are currently in effect (treated the same as LOG=?, LIMITS=? and FILTER=?).

Alarm Filtering Commands

- Enter **ALARM FILTER=(A|S|P,*id*1(A|S|P|FC|FU),...,*id*n(A|S|P|FC|FU))** to set alarm IDs and actions.

  The first FILTER suboperand specifies the default action to be taken when an alarm is issued.

  A (Allow) specifies that any alarm that *is not identified* in the filter list *is to be* displayed.

  S (Suppress) specifies that any alarm that *is not identified* in the filter list *is not to be* displayed.

  P (Purge) specifies that any alarm that *is not identified* in the filter list *is not to be* displayed *or* written to SYSPRINT.

  If no value is specified for the first FILTER suboperand (FILTER=(,*id*1,...)), A is assumed. For more information on HNAS alarm filtering support, please refer to the description of the ALRMFLTR= operand on the BUILD definition statement which is described in Chapter 4 of the HNAS Guide and Reference manual.

  Alarm filter identifiers ***id1...,idn...*** are given in the form '**NAS*xxxxs***' where '**NAS**' is the HNAS message identifier, '***xxxx***' is the message number and '***s***' is the severity code. You may enter up to 16 different alarm filter values. Wildcards (*) are accepted.

  The filters you enter remain in effect until changed or until the console session terminates. For more information on HNAS alarm messages, please refer to sections Configuration Messages and Alert Messages in the HNAS Messages and Codes book.

  You may also suffix an alarm filter identifier with **(A)**, **(S)**, **(P)**, **(FC)** or **(FU)** to override the default action. Note that the **A|S|P|FC|FU** suffix must be enclosed in parenthesis.

  Note: The *first* FILTER= operand entry is the default action and will be used if a match or no match is found with the ***id*i** values in the list. To override the default action, you must specify an override value in parentheses after each ***id*i** value. If no override action is given, the default action is used.

  For example, specifying ALARM FILTER=(P,NAS3799I) is the same as specifying ALARM FILTER=(P,NAS3799I(P)).

  (A) (Allow) specifies that the filtered alarm message is to be allowed to go to SYSCONS (subject to the current SHOW state) regardless of the default action.

  (S) (Suppress) specifies that the filtered alarm message is to be suppressed from SYSCONS regardless of the default action.

(P) (Purge) specifies that the filtered alarm message is to be purged from SYSCONS and SYSPRINT regardless of the default action.

(FC) (Force Conditionally) specifies that the filtered alarm message is to be sent to SYSCONS subject to the current SHOW state.

For example, if ALRMFLTR=(...,NAS2020I(FC),...) is specified, the NAS2020I alarm messages will be sent to SYSCONS except when SHOWOFF is effect.

(FU) (Force Unconditionally) specifies that the filtered alarm message is to be sent to SYSCONS regardless of the current SHOW state.

For example, if ALRMFLTR=(...,NAS0299I(FU),...) is specified, the NAS0299I alarm messages will be sent to SYSCONS even when SHOWOFF is in effect.

Note: (FC) and (FU) message ID suffix support was introduced into 240 via APAR 2400021.

To purge all but the NAS3799I message, specify ALARM FILTER=(P,NAS3799I(A)) or ALARM FILTER=(P,NAS3799I(S)). The (A) causes the NAS3799I message to be written to both SYSCONS and SYSPRINT. The (S) restricts the message to SYSPRINT only. All other messages will be purged from SYSCONS and SYSPRINT because the *first* FILTER operand (default disposition) was P (PURGE). We recommend using A (ALLOW) as the default disposition and specify (P) or (S) as appropriate for the ***id*i** values in the list. If P is specified as the default disposition, important error messages could be lost.

Note: The show error console command function **SHOW ERR** (start parameter **SHOWERR**) currently affects the processing of alarm message filtering because the SHOW processing is executed before alarm filter processing. This means that when SHOWERR is in effect (enabled), informational alarms will only be filtered by the first ALRMFLTR= operand (the default disposition). In this case informational alarms will not be written to SYSCONS but will be written to SYSPRINT. To purge information alarms from both SYSCONS and SYSPRINT when SHOWERR is active, specify FILTER=(P,NAS*(A)). Another way to do this is to specify **SHOW ON** (start parameter **SHOWON**) and FILTER=(A,NAS****I(P)). SHOWON makes informational alarms subject to full alarm filtering not just the default disposition.

Note: The initial implementation of the HNAS ALARM console command processor replaced existing FILTER= values instead of appending values to the ALRMFLTR= array. This restriction forced users to reenter existing alarm list entries (previously specified on the BUILD ALRMFLTR=) so that they could add new message identifiers (***id*i**) via the console.

The HNAS ALARM console command processor has been modified to append FILTER= values to the existing ALRMFLTR= array rather than resetting the array first. This eliminates the need to enter an entire list of alarm identifiers in order to add just one entry to the end of the list. For example, if ALRMFLTR=(ALLOW,NAS2***I(P)) was specified on the BUILD definition statement in the CDF and if ALARM FILTER=(NAS3701W(P)) is then issued as a console command, the ALARM FILTER=? command would produce the following display:

```
ALRMFLTR=ALLOW
         NAS2***I(P)
```

```
NAS3701W(P)
```

- Enter **ALARM FILTER=*** to force the alarm filter array to be cleared. The array count is set to zero, all alarm filter IDs are set to null values and the default disposition is set to ALLOW.

  If you wish to clear the ALRMFLTR= array before entering new values, enter ALARM FILTER=* to clear the entries followed by ALARM FILTER=(A|S|P,*idlist*) for the new filter entries.

- Enter **ALARM FILTER=?** to display the alarm filter parameters that are currently in effect. The following is example of the information produced by the ALARM FILTER=? command.

```
ALRMFLTR=SUPPRESS
          NAS3****(S)
          NAS*****(A)
```

Note: For the alarm filter list above, messages that start with 'NAS3' will not be displayed even though NAS*****(A) is also specified. This is because HNAS processes the alarm filter list left to right (from the first to the last entry) and NAS3****(S) appears before NAS*****(A) in the list.

Note: Informational monitor TAP messages of the form **NAS*xxxx*M** and informational trace messages of the form **NAS*xxxx*T** cannot be filtered using the ALRMFLTR= operand. NAS*xxxx*M messages can be suppressed via the MON TAP ALLOFF command while NAS*xxxx*T messages can be suppressed via the TRCPRNT OFF command. Monitor TAP information processing differs from trace information processing in that the former does not also log data in the internal trace table. When MON TAP ALLOFF is in effect, no trace data is logged unless TRCPCE is active for the TAP PCE. Conversely, when TRCPRNT OFF is in effect, trace data is still logged in the internal trace table because TRCLU ALLON, TRCVC ALLON, TRCMCH ALLON and TRCMCHX ALLON are set by default at HNAS startup time.

Note: You may enter **ALRMFLTR=** or **FILTER=** as an ALARM command operand. These keywords are treated the same and their list values are processed identically.

Alarm Limiting Commands

(new for V2R2M0)

Alarm messages that are allowed are subject to a second level of filtering based on their severity in a 30-second interval. In other words, HNAS will only allow a certain number of messages to be displayed every 30-seconds. This was done to prevent the WTO buffer pool from becoming depleted. The following list provides the **default limits** for allowed alarms as a function of their severity.

20 - maximum number of **I**nformational messages allowed during interval.
10 - maximum number of **D**efault messages allowed during interval.
20 - maximum number of user **A**ction messages allowed during interval.
25 - maximum number of **W**arning messages allowed during interval.
30 - maximum number of **E**rror messages allowed during interval.
30 - maximum number of **S**evere error messages allowed during interval.
10 - maximum number of all **M**iscellaneous messages allowed during interval.

The default alarm limits and the display interval can be modified using the LIMITS= operand of the ALARM command (see below).

- Enter **ALARM LIMITS=(*interval*,*ilmt*,*dlmt*,*almt*,*wlmt*,*elmt*,*slmt*,*mlmt*)** to set alarm limits.

  The first LIMITS= suboperand specifies the time ***interval*** in seconds in which the limits will be counted. The balance of the LIMITS= parameter is used to specify the maximum counts for each message severity code. The values are positional and each may be specified as a decimal number between 0 and 255. A value of zero is treated as no limit. To leave a limit at its current value, you may bypass it in the list by using 2 consecutive commas (,,).

  For example, entering ALARM LIMITS=(60,75,20,40,50,60,60,20) will change the count interval and limit values as follows:

  60 - count interval.
  75 - maximum number of **I**nformational messages allowed during interval.
  20 - maximum number of **D**efault messages allowed during interval.
  40 - maximum number of user **A**ction messages allowed during interval.
  50 - maximum number of **W**arning messages allowed during interval.
  60 - maximum number of **E**rror messages allowed during interval.
  60 - maximum number of **S**evere error messages allowed during interval.
  20 - maximum number of all **M**iscellaneous messages allowed during interval.

  When the count for a particular message severity code reaches its limit value, subsequent messages of the same severity will not be displayed during the current interval. When the interval expires, the counts are reset so that alarms may again be displayed during the next interval. Note that the **LIMITS=** parameter of the **ALARM** command can be used to set all **ALARMLMTS=** operand values except the log count (***lcnt***). The ***lcnt*** value remains fixed for the active configuration. For more information on HNAS alarm limiting support, please refer to the description of the **ALRMLMTS=** operand on the BUILD definition statement which is described in Chapter 4 of the HNAS Guide and Reference manual.

- Enter **ALARM LIMITS=*** to force the alarm limits array to be cleared. All defaults are reset (zero set as all limits which means no limit enforced).

- Enter **ALARM LIMITS=?** to display the alarm limit parameters that are currently in effect. The following is example of the information produced by the ALARM LIMITS=? command.

```
ALRMLMTS=30 (TIMER)
         20 (INFO)
         10 (DEFAULT)
         20 (ACTION)
         25 (WARNING)
         30 (ERROR)
         30 (SEVERE)
         10 (MISC)
```

  Note: You may enter **ALRMLMTS=** or **LIMITS=** as an ALARM command operand. These keywords are treated the same and their list values are processed identically.

Alarm Logging Commands

(new for V2R2M0)

- Enter **ALARM LOG=CLEAR** to remove all entries from the alarm logging table.

- Enter **ALARM LOG=RESET** to reset all the counters for all existing entries in the alarm logging table.

  Note: The HNAS alarm logging table is allocated based on the ***lcnt*** suboperand of the **ALRMLMTS=** operand on the BUILD definition statement. If an ***lcnt*** value is specified, alarms are logged unconditionally even if they are not displayed because of filtering or limiting. Entries in the alarm logging table are sorted by alarm ID. For example, NAS2401W appears before NAS7703W in the table. In addition to the alarm ID, each entry in the alarm logging table contains a count indicating the number of times the alarm message was issued as well as a timestamp indicating the last time the alarm was issued.

- Enter **ALARM LOG=?** to display the alarm logging table. The following is example of the information produced by the ALARM LOG=? command.

```
ALARM_ID  TOTAL_CT    LOG_DATE  LOG_TIME   (0013/0256)
NAS0001I  00000001  2006/03/02  11:22:12
NAS1700I  00000001  2006/03/02  11:22:09
NAS1709I  00000003  2006/03/02  11:22:09
NAS1730I  00000007  2006/03/02  11:22:09
NAS2010I  00000001  2006/03/02  11:22:11
NAS2020I  00000001  2006/03/02  11:22:11
NAS2030I  00000001  2006/03/02  11:22:10
NAS2050I  00000001  2006/03/02  11:22:11
NAS2200I  00000001  2006/03/02  11:22:11
NAS2230I  00000001  2006/03/02  11:22:11
NAS2240I  00000001  2006/03/02  11:22:11
NAS3701W  00000002  2006/03/02  11:22:16
NAS3798I  00000002  2006/03/02  11:22:16
```

  Note: The counts in parentheses (**0013/0256**) on the header line above represent the current number of entries in the alarm logging table and the maximum number entries (***lcnt*** from ALRMLMTS= operand) that the table can accommodate. The table is ordered by Alarm ID so that the **ALARM_ID** column shows alarms in collating sequence. The **TOTAL_CT** column represents the total number of times that the given alarm was issued since HNAS was loaded or the alarm logging table was reset. The **LOG_DATE** and **LOG_TIME** columns represent the most recent occurrence for the given alarm.

  Note: The **ALARM LOG=?** command is executed automatically when HNAS is SHUTDOWN using the Q (QUIT) command. This is done unconditionally for any QUIT command follower. If QE *ddname* is entered, the ALARM LOG=? command is executed before the command list identified by *ddname*.

Note: You may enter **ALRMLOG=** or **LOG=** as an ALARM command operand. These keywords are treated the same and their values are processed identically.

Note: Specific alarm messages that are flushed when **NAS0111W** nnnn ALARM LOST events occur are not included in any of the **ALARM LOG** bucket counts although occurrences of the NAS0111W messages are.

Alarm Format Control Commands

(added for V2RM0)

- Enter **ALARM MSGTXT=SHORT** to force alarm messages to be displayed on a single line (short format mode). This sets the BUILD OPTIONS=ALRMSGTXT= operand value to SHORT.

- Enter **ALARM MSGTXT=LONG** to force alarm messages to be displayed on a multiple lines (long format mode). This sets the BUILD OPTIONS=ALRMSGTXT= operand value to LONG.

  ALARM MSGTXT={SHORT|LONG} allows you to select the short or long format for alarm messages. Heretofore, only long format alarm messages were provided which can produce multiple lines of output for each alarm. Short format alarm messages consolidate information on a single line. This option applies to all alarm messages including informational alarms.

  For example, when ALARM MSGTXT=LONG is in effect, the NAS2200I message is displayed as follows:

```
NAS2200I SERVER=010.117.056.171(01998) SOCKID=0000 PCEID=0009 NAME=LXOT
NAS2200I SOCKET ASSIGNED BY SERVER, SOCCNT=00001 SOCLMT=02000
```

  When ALARM MSGTXT=SHORT is in effect, this same message is displayed as follows:

```
NAS2200I SERVER=010.117.056.171(01998) NAME=LXOT SOCKET ASSIGNED BY SERVER, ...
         ... SOCCNT=00001 SOCLMT=02000
```

  For additional information on short and long alarm message processing, please refer to the description of the BUILD OPTIONS=ALRMSGTXT= operand in Chapter 4 of the HNAS Guide and Reference manual.

- Enter **ALARM MSGTXT=?** to display the current value for the BUILD OPTION=ALRMSGTXT= operand.

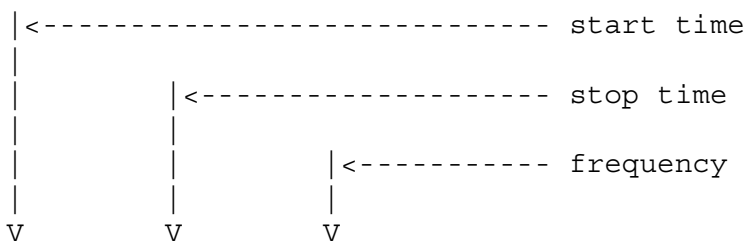
  Note: You may enter **ALRMSGTXT=** or **MSGTXT=** as an ALARM command operand. These keywords are treated the same and their list values are processed identically.

  Note: ALARM MSGTXT={?|SHORT|LONG} command support was introduced into 240 as Enhancement APAR 2400068.

Alarm Heartbeat Commands

(apar/added for V2R4M0)

- Enter **ALARM PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)** to update the alarm PULSE parameters. The syntax meaning is as follows:




```
ALARM PULSE=(hh:mm:ss,hh:mm:ss,seconds)
```

Note: Prior to updating the PULSE= operand values that are currently in effect, the values are copied into a temporary workarea. This allows you to modify one or all PULSE= suboperands before the values are returned to the permanent save area. For example if you only want to change the PULSE frequency, enter ALARM PULSE=(,,*newseconds*). The start and end times will remain the same.

HNAS will issue the following message at the frequency given by the *seconds* value within the interval specified by the start and stop times:

```
NAS0299I HNAS PULSE TAKEN AT hh:mm:ss.uu ON yyyy/mm/dd
```

If the stop time is less than the start time, the interval wraps through midnight. If the start time and end time are equal, pulsing will be continuous.

The purpose of the message is to provide an indication that HNAS is being dispatched on a regular basis. The SYSPRINT log will contain the NAS0299I message. The message can also be sent to SYSCONS if ALRMFLTR=(...,NAS0299I(FU),...) is also specified on the BUILD definition statement or if the ALARM FILTER=(...,NAS0299I(FU),...) console command is issued.

- Enter **ALARM PULSE=*** to force the alarm PULSE parameters to be cleared. HNAS heartbeat processing is disabled.

  If you wish to clear the PULSE= parameters before entering new values, enter ALARM PULSE=* to clear the entries followed by ALARM PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*) for the new PULSE parameters.

- Enter **ALARM PULSE=?** to display the alarm PULSE parameters that are currently in effect.

  Note: PULSE= operand support was introduced into 240 via APAR 2400021.

CID= Modifier - Set Terminal Session LUB Identifier

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
CID=*mincid*[-*maxcid*] **CID=?**	***command***	**[CID=*mincid*[-*maxcid*]]**	**C**	**240**

This modifier is used to select specific terminal LU(s) for the commands that follow. HNAS allocates a LUB for each REMOTE terminal LU connection. The **CID=** value serves as an identifier for HNAS terminal LU resources. For more information on LU identifiers, refer to page CONS-36 of this document.

Note: The CID= modifier is used by the DLU and TRCLU commands.

- If CID= (null) is specified, the command operates on all LUBs unless noted otherwise.
- If CID=0 is specified (CID=0-0 is not permitted), the command also operates on all LUBs unless noted otherwise. For all commands, CID= (null) and CID=0 are treated identically.
- If a CID= value is specified as a range (e.g., CID=3-10), the command operates on the selected LUBs.
- If a CID= value is specified as a single element (e.g., CID=3) or as a range where the extents are the same (e.g., CID=3-3), the command operates on the selected single LUB.
- If an invalid CID= value is specified, the balance of the command line is ignored.
- If a CID= value is specified, it must be entered as a hexadecimal number.

  CID=*mincid*[-*maxcid*] values can range from 1 to FFFFFFFE.

- Enter **CID=?** to display the permanent CID= value.

CLDADDR= Modifier - Set Called DTE Address

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
CLDADDR=*ddd...ddd* **CLDADDR=?**	***command***	**[CLDADDR=*ddd...ddd*]**	**N** **C**	**114** **240**

This modifier is used to select a specific ***called*** DTE address for the TRCADDR and PING commands. The TRCADDR command is used to start or stop tracing for a specific **local** DTE regardless of virtual circuit or TCP socket that it uses for host access.

- You may enter up to 15 decimal digits for the ***ddd...ddd*** value.
- If a null CLDADDR= value is entered, the remembered CLDADDR= modifier is reset.
- Enter **CLDADDR=?** to display the permanent CLDADDR= value.

CLGADDR= Modifier - Set Calling DTE Address

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
CLGADDR=*ddd...ddd* **CLGADDR=?**	***command***	**[CLGADDR=*ddd...ddd*]**	**N** **C**	**114** **240**

This modifier is used to select a specific ***calling*** DTE address for the TRCADDR command. The TRCADDR command is used to start or stop tracing for a specific **remote** DTE regardless of virtual circuit or TCP socket that it uses for host access.

- You may enter up to 15 decimal digits for the ***ddd...ddd*** value.
- If a null CLGADDR= value is entered, the remembered CLGADDR= parameter is reset.
- Enter **CLGADDR=?** to display the permanent CLGADDR= value.

DADDR Command - Display TRCADDR Addresses

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DADDR	**[ALL]** |	**N**	**114**
		[CLDADDR[=]]		
		[CLGADDR[=]]		
		[IPADDR[=]]		

This command will display the contents of the TRCADDR address table. The TRCADDR address table is used to restrict HNAS event tracing to specific resources independent of the physical path over which they are connected. The TRCADDR address table can contain up to 16 ***called*** DTE addresses and/or 16 ***calling*** DTE addresses and/or 16 **TCP sockets**. Entries are added or removed from the TRCADDR address table using the **TRCADDR** command. For more information, please refer to page CONS-247 of this document.

If no resource address list is specified, **ALL** is assumed. Note that entering **DADDR** or **DADDR ALL** is equivalent to entering **DADDR CLDADDR= CLGADDR= IPADDR=**.

To summarize, you may enter a DADDR parameter that requests a specific action to be performed as follows:

DADDR Action Matrix

PARAMETER	ACTION
ALL or omitted	**Display all addresses in table.**
CLDADDR=	**Display all *called* DTE addresses in table.**
CLGADDR=	**Display all *calling* DTE addresses in table.**
IPADDR=	**Display all TCP sockets (IPADDR/PORT) in table.**

The following is an example of the heading and information produced by the **DADDR** command.

```
REMOTE  RESOURCE              TYPE
|                             |
|                             |<- Address Type
|                                 CLDADDR|CLGADDR|IPADDR
|
|<- Address Value
```

The following display is produced when **DADDR** is entered as the last command of following command sequence:

```
CLGADDR=12345678901234 TRCADDR ADD
CLDADDR=1024999 TRCADDR ADD
IPADDR=1.2.3.4(1096) TRCADDR ADD
DADDR

REMOTE RESOURCE           TYPE
12345678901234          CLGADDR
1024999                 CLDADDR
001.002.003.004(01096)   IPADDR
```

D or DMEM Command - Display Memory

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	D[MEM]	***address*** **[+*offset*]** **[{*length*|80}]**		

This command will display HNAS storage areas.

You may enter an absolute ***address*** or a relative address as an offset from an established base (**+ *offset***). A new base is set when the ***address*** and **+ *offset*** parameters are entered together. For example, 'D 6500 + 20 40' will display from address 6520 for 40 hexadecimal bytes and set the address base to 6500. The address base is initially set to the NASMAIN load point.

You may enter a display ***length*** (number of hexadecimal bytes) between 1 and FFF0 (65520).

If a ***length*** value is not provided, a default length of **80** hexadecimal bytes is used.

The following display is produced when **D 20000 90** is entered.

```
 ADDRESS  OFFSET-0  OFFSET-4  OFFSET-8  OFFSET-C  EBCDIC TEXT DATA
00020000 47F0F02C 0026D5C1 E2D4C1C9 D540E5F2  00...NASMAIN V2
00020010 D9F2D4F0 4040E5D4 40404040 4040F1F1 R2M0  VM      11
00020020 61F1F761 F0F20000 0000145C 90ECD00C /17/02.....*  }.
00020030 18AF182D 189158C0 AB284130 06C05030 . ...j { . ..{&.
00020040 C0081803 4510A048 0A0A5010 C00C4163 {... .  ..&.{.
00020050 10005060 C01018D1 18011813 1B441B55 ..&-{..J..... .
00020060 0E041832 184D4150 00185040 30085030 .....( &..& ..&.
00020070 40041834 41404048 4650A06A 430C0092  ...     & ] ..k
00020080 41F0000F 140F0700 4D10A0AE 001E8000  0......(.  .. .
```

Note: If the DMEM command references a memory address that HNAS is not authorized to display (the address is invalid for the machine configuration), the following error message will be displayed:

```
NASC611W DISPLAY MEMORY ADDRESSING EXCEPTION, CONSOLE COMMAND ABORTED
```

DLCL Command - Display LOCAL Configuration

(new for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[LNM=*lclname*]	**DLCL**	**[*lclname*]**	**C**	**230**
		[ALL] |	**N**	**220**
		[INIT[=]]	**N**	**230**
		[IPADDR[=]]		
		[OPTIONS[=]]		
		[PORT[=]]		
		[RTEIN[=]]	**C**	**240**
		[RTEOUT[=]]		
		[SOCLMT[=]]		
		[TYPE[=]]		

This command will display the operands that were specified in the CDF or modified later via the MLCL command for the LOCAL definition statement identified by the **LNM=** modifier. If no **LNM=** value is set, the list of all LOCAL definition statements is displayed.

To summarize, you may enter a list of LOCAL definition statement operands that you want to display as follows:

DLCL Action Matrix

PARAMETER	ACTION	XTP	XOT
ALL or omitted	**Display value for all LOCAL operands.**		
INIT=	**Display value for INIT= operand.**		
IPADDR=	**Display value for IPADDR= operand.**		
OPTIONS=	**Display value for OPTIONS= operand.**		
PORT=	**Display value for PORT= operand.**		
RTEIN=	**Display value for RTEIN= operand.**	**I**	
RTEOUT=	**Display value for RTEOUT= operand.**	**I**	
SOCLMT=	**Display value for SOCLMT= operand.**		

PARAMETER	ACTION	X T P	X O T
TYPE=	**Display value for TYPE= operand.**		

Legend: For LOCAL *type* column: I=> Invalid, blank=> supported (see Console Command Parameter Action Matrix Keys description on page CONS-21 for additional information).

If no operand list is specified, **ALL** is assumed. Note that entering **DLCL [ALL]** is equivalent to entering **DLCL IPADDR= OPTIONS= ... TYPE=**.

Certain keywords may specify a REMOTE name (e.g., RTEOUT=*rmtname*), in which case only the list entry that corresponds to the named REMOTEs will be displayed (see example below).

For keywords on a TYPE=XOT LOCAL definition statement, NONE will be displayed when NONE was coded for the operand in the CDF (e.g., RTEOUT=NONE). If the operand was omitted in the CDF, a null value will be displayed (e.g., RTEOUT=).

The following is an example of the heading and information produced by the **DLCL** command.

```
LCLNAME  OPERAND  TYPE
|        |        |
|        |        |<- LOCAL type
|        |
|        |<- Operand/Value or
|           LOCAL address
|
|<- LOCAL name
```

The following display is produced when **DLCL LNM=** is entered (no LOCAL name given).

```
LCLNAME  OPERAND  TYPE
LXTP     00075CC4 XTP
LXOT     00075D10 XOT
```

The following display is produced when **DLCL** [**LNM=**]**LXOT** is entered.

```
LCLNAME  OPERAND  TYPE
LXOT     00075D10 XOT
         INIT=ACTIVE
              DELAYTIME=5
              RETRYLMT=0
         IPADDR=010.117.056.221
         OPTIONS=
```

```
          PORT=01998
          RTEIN=MCH1/2036T
                MCH1/9999T
                MCH1/661T
          RTEOUT=R2CNOT1/1036T
                 R2CNOT2/10369999T/NYC USER #1
          SOCLMT=01000
          TYPE=XOT
```

The following display is produced when **LNM=LXOT DLCL IPADDR= RTEOUT=** is entered.

```
LCLNAME   OPERAND  TYPE
LXOT      00075D10 XOT
          IPADDR=010.117.056.221
          RTEOUT=R2CNOT1/1036T
                 R2CNOT2/10369999T/NYC USER #1
```

The following display is produced when **LNM=LXOT DLCL RTEOUT=R2CNOT2** is entered.

```
LCLNAME   OPERAND  TYPE
LXOT      00075D10 XOT
          RTEOUT=R2CNOT2/10369999T/NYC USER #1
```

DLP Command - Display HNAS Load Point

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DLP			

This command will display the HNAS load point in the HNAS address space.

The following display is produced when **DLP** is entered.

```
NAS LOADPOINT=00007000
```

DLU Command - Display Logical Unit

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

for a specific LU,

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[LUNM=*sluname*]	**DLU**	**[*sluname*]**	**C**	**230**
		[{SHOWNAME[{=RTR| **=MCH|** **=SPU}]|** **SHOWADDR}]**	**N** **C**	**230** **230**
		[{SHOWBNDAPPL| **SHOWUNBAPPL}]**	**A**	**230**
		[{SHOWSESSINIT|SHOWCID}]	A	240
		[{FMT1|FMT2}]		

or for a collection of LUs on a specific MCH or SPU

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM={*mchname*|*spuname*}]	**DLU**	**[{*mchname*|*spuname*}]**	**C**	**230**
[CID=*mincid*[-*maxcid*]]		**[{SHOWNAME[{=RTR|** **=MCH|** **=SPU}]|** **SHOWADDR}]**	**N** **C**	**230** **230**
		[{SHOWBNDAPPL| **SHOWUNBAPPL}]**	**A**	**230**
		[{SHOWSESSINIT|SHOWCID}]	N	240
		[{FMT1|FMT2}]		
		[{CONNECTED|NONIDLE}]	N	240
		[STATE=BOUND| **CONNECTED** **CLSDST|** **IDLE|** **NONIDLE|** **OFFLINE|** **OPNDST|** **PLUBIND|** **PLUUNBIND|** **REQSESS|** **SLUUNBIND]**	**N** **N** **N** **N**	**220** **240** **240** **240**
		[TRACE={OFF|ON}]	**N**	**230**

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[TYPE={GCTL|LLC4}| GDAT| IDLE| {PAD|LLC5}| {PCNE|LLC0}| {QLLC|LLC3}]	**N**	**220**

or for a collection of LUs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**DLU**	**{SHOWNAME[{=RTR| =MCH| =SPU}]| SHOWADDR}]**	**N** **C**	**230** **230**
[IFN=*minifn*[-*maxifn*]]		**[{SHOWBNDAPPL| SHOWUNBAPPL}]**	**A**	**230**
[CID=*mincid*[-*maxcid*]]		**[{SHOWSESSINIT|SHOWCID}]**	N	240
		[{FMT1|FMT2}]		
		[{CONNECTED|NONIDLE}]	N	240
		[ALLID]		
		[ALLIFN]		
		[ALLCID]		
		[PROT={XOT|XTP}]		
		[STATE=BOUND| **CONNECTED** **CLSDST|** **IDLE|** **NONIDLE|** **OFFLINE|** **OPNDST|** **PLUBIND|** **PLUUNBIND|** **REQSESS|** **SLUUNBIND]**	**N** **N** **N** **N**	**220** **240** **240** **240**
		[TRACE={OFF|ON}]	**N**	**230**
		[TYPE={GCTL|LLC4}| GDAT| IDLE| {PAD|LLC5}| {PCNE|LLC0}| {QLLC|LLC3}]	**N**	**220**

This command will display HNAS SLU state information. If only the HEADER line is displayed after the command is entered it indicates that nothing matched the display criteria as requested by the command parameters entered or those currently in effect.

- When **LUNM=*sluname*** is specified, it overrides the **ID=**, **IFN=**, **CID=** and **RNM=** values restricting DLU processing to the named SLU only.

- When **RNM={*mchname*|*spuname*}** is specified, it overrides the **ID=** and **IFN=** values restricting DLU processing to the named MCH or SPU for the specified **CID=** values. DLU processing is for the SLUs on the named MCH only.

- When **CID=*mincid*[-*maxcid*]** is specified, DLU processing is restricted to the selected SLUs on the selected MCHs. If **CID=0** is specified, DLU processing is for all SLUs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), DLU processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, DLU processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, DLU processing is restricted to the selected PCEs. If **ID=0** is specified, DLU processing is for all PCEs

- The **SHOWNAME** or **SHOWNAME=RTR** operand (the default) requests that the name of the TYPE=XOT|XTP REMOTE definition statement associated with the SLU is to be displayed in the RTRNAME/ADDRESS column. This REMOTE identifies the router that is supporting the SLU/VC connection. If the SLU does not have an active VC connection, no name is displayed.

- The **SHOWNAME=MCH** operand requests that the name of the TYPE=MCH|XTP REMOTE definition statement associated with the SLU is to be displayed in the MCHNAME/ADDRESS column. This REMOTE identifies the logical MCH that is supporting the SLU/VC connection. If the SLU does not have an active VC connection, no name is displayed.

- The **SHOWNAME=SPU** operand requests that the name of the TYPE=SPU REMOTE definition statement associated with the SLU is to be displayed in the SPUNAME/ADDRESS column. This REMOTE identifies the SPU that is supporting the SLU connection. If the SLU is not defined in the LUNAME= operand on a TYPE=SPU REMOTE definition statement, no name is displayed.

- The **SHOWADDR** operand requests that the memory address of the SLU control block (LUB) is to be displayed in the *xxx*NAME/ADDRESS column.

- The **SHOWBNDAPPL** operand (the default) requests that the name of the PLU currently in session with the SLU is to be displayed in the PLUNAME column. If the SLU is not bound, no name is displayed.

- The **SHOWUNBAPPL** operand requests that the name of the PLU currently or previously in session with the SLU is to be displayed in the PLUNAME column. This name is displayed even if the SLU is no longer bound. If the SLU has never been bound, no name is displayed.

- The **SHOWSESSINIT** operand (the default) requests that the direction of the session initiation is to be displayed in the SESSINIT/CID column. If the SLU session was established via an inbound or outbound call, INBOUND or OUTBOUND is displayed. If the SLU is not connected, blanks are displayed.

- The **SHOWCID** operand requests that the VTAM connection identifier is to be displayed in the SESSINIT/CID column.

- The **FMT1** and **FMT2** operands (for format 1 and 2) control the type of information that is displayed. Currently, **FMT1=SHOWNAME+SHOWBNDAPPL+SHOWSESSINIT** (the default) and **FMT2=SHOWADDR+SHOWUNBAPPL+SHOWCID**.

  Note: The SHOW*xxxx* and FMT*x* operands are processed in the order specified. The last value you specify is the winner. For example, if you enter SHOWADDR FMT1, SHOWNAME will override SHOWADDR because FMT1 includes SHOWNAME.

- The **CONNECTED** operand requests that only those SLUs that have an active VC connection are to be displayed. **VCACT** is a alternate name for CONNECTED. CONNECTED or VCACT forces ID=IFN=CID=0 and RNM=LUNM= null so that all defined SLUs are tested for an active VC connection.

- The **NONIDLE** operand requests that only those SLUs that have a non-idle SNA state are to be displayed. The SLU may or may not also have an active VC connection **LUACT** is a alternate name for NONIDLE. NONIDLE or LUACT forces ID=IFN=CID=0 and RNM=LUNM= null so that all defined SLUs are tested for a non-idle SNA state.

- The **ALLID** parameter is treated as though RNM= omitted, LUNM= omitted and ID=0 were in effect even if they are not. This means that all PCEs are scanned for their MCH components.

- The **ALLIFN** parameter is treated as though IFN=0 was in effect even if it is not. This means that all MCHs are scanned for their SLU components.

- The **ALLCID** parameter is treated as though CID=0 was in effect even if it is not. This means that all SLUs are displayed for each MCH subject to filtering by the **PROT=**, **STATE=**, **TRACE=** and/or **TYPE=** operands.

- The **TYPE=** parameter is used to restrict the display to a specific SLU type.

- The **PROT=** parameter is used to restrict the display to a specific protocol for **TYPE=TCP** PCEs over which the SLUs communicate.

- The **STATE=** parameter is used to restrict the display to a specific SLU state.

  Note: **STATE=CONNECTED|NONIDLE** is treated the same as the CONNECTED|NONIDLE arguments except that the command modifiers are not forced to their null values. This means that only those SLUs selected by the provided command modifiers are tested for an active VC connection or non-idle SNA state.

  Note: When **STATE=BOUND** is specified, STAT column will show secondary bound state which can be SDT, CLR, DTA or SD rather than BND.

Note: When **STATE=OFFLINE** is specified, only those SLUs that have been varied offline with the console VARY command will be displayed subject to the command modifiers in effect.

- The **TRACE=** parameter is used to restrict the display to SLUs being traced (**ON**) or not being traced (**OFF**).

The following is an example of the heading and information produced by the **DLU** command.

```
                          SPUNAME
                          MCHNAME
     IFN| CID             RTRNAME
MPID RPID SSESINIT VCN    ADDRESS LUOPT LUST LUTO TYPE SLUNAME PLUNAME
|     |     |          |      |       |      |     |     |       |
|     |     |          |      |       |      |     |     |       |<-PLU name
|     |     |          |      |       |      |     |     |          (APPL)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- HNAS SLU name
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- SLU type (8)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- Active SLU timer (7)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- SLU state (6)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- SLU runtime options (5)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- REMOTE name or LUB address (4)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- SLU virtual circuit number when a network
|     |     |          |      |       |      |     |     |   VC connection is active, blank otherwise (3)
|     |     |          |      |       |      |     |     |   Note: For PVCs, the first digit is a 'P'
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- Session initiation direction (INBOUND or OUTBOUND) or
|     |     |          |      |       |      |     |     |   the SLU connection identifier when SLU is bound, zero
|     |     |          |      |       |      |     |     |   otherwise (2)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- MCH interface number for an XTP router connection or the PCE
|     |     |          |      |       |      |     |     |   ID of the TCP/IP component for an XOT router connection that
|     |     |          |      |       |      |     |     |   owns the SLU/VC session (1)
|     |     |          |      |       |      |     |     |
|     |     |          |      |       |      |     |     |<- PCE ID for the MCH that owns the named SLU
```

Note: MPID and IFN|RPID header display change was introduced into 240 with Enhancement APAR 2400014.

The following display is produced when **DLU ID=8 IFN=3 CID=7-9 SHOWNAME=MCH** is entered.

```
     IFN|
MPID RPID SESSINIT VCN  MCHNAME  LUOPT LUST LUTO TYPE SLUNAME  PLUNAME
0008 0003 INBOUND  P001 MCH1     LDONP  DTA       PAD R1M3P001 A06TSO
0008 0003 OUTBOUND P002 MCH1     LDONP  DTA       PAD R1M3P002 A06TSO
```

The following display is produced when **DLU ID=8 IFN=3 CID=7-9 FMT2** is entered.

```
     IFN|
MPID RPID CID      VCN  ADDRESS  LUOPT LUST LUTO TYPE SLUNAME  PLUNAME
0008 0003 00000007 P001 0009D220 LDONP  DTA       PAD R1M3P001 A06TSO
0008 0003 00000009 P002 0009D720 LDONP  DTA       PAD R1M3P002 A06TSO
```

In the previous example, no listing is given for CID=8 because the SLU is idle, that is, CID=8 is not assigned.

The following display is produced when **DLU BPM FMT2** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
     IFN|
MPID RPID CID      VCN  ADDRESS  LUOPT LUST LUTO TYPE SLUNAME  PLUNAME
0008 0003 00000000      00097968 LD NP IDLE      PCNE R1M30005
0008 0003 00000000      0009C8C8 LD NP IDLE       PAD R1M35020
0008 0003 00000007 P001 0009D220 LDONP  DTA       PAD R1M3P001 A06TSO
0008 0003 00000009 P002 0009D720 LDONP  DTA       PAD R1M3P002 A06TSO
0008 0003 00000000 P003 0009DC20 LD GP IDLE      GDAT R1M3P003
0008 0003 00000000 P004 0009E120 LD GP IDLE      GDAT R1M3P004
0008 0003 00000000 P005 0009E620 LD GP IDLE      GDAT R1M3P005
0008 0004 00000000      0009FA88 LD NP IDLE      PCNE R1M40004
0008 0004 00000000      000A0A68 LD NP IDLE       PAD R1M45004
000D 010A 00000000      000A2820 LD NP IDLE      PCNE MCH10005
000D 010B 00000000      000A7780 LD NP IDLE       PAD MCH15000
000D 010C 00000000 0001 000A80D8 LD NP IDLE       PAD MCH15001 A06TSO
000D 010D 00000000 0002 000A85D8 LD NP IDLE       PAD MCH15002 A06TSO
000D 010E 00000000 0003 000A8AD8 LD G  IDLE      GDAT MCH14003
000D 010F 00000000 0004 000A8FD8 LD G  IDLE      GDAT MCH14004
000D 0110 00000000 0005 000A94D8 LD G  IDLE      GDAT MCH14005
```

In the previous example, all SLUs for all MCHs are listed including those that are idle (CID=0) because no CID modifier was supplied for the DLU command.

The following display is produced when **DLU [RNM=]MCH1 FMT2** is entered.

```
     IFN|
MPID RPID CID      VCN  ADDRESS  LUOPT LUST LUTO TYPE SLUNAME  PLUNAME
000D 010A 00000000      000A2820 LD NP IDLE      PCNE MCH10005
000D 010B 00000000      000A7780 LD NP IDLE       PAD MCH15000
000D 010C 00000000 0001 000A80D8 LD NP IDLE       PAD MCH15001 A06TSO
000D 010D 00000000 0002 000A85D8 LD NP IDLE       PAD MCH15002 A06TSO
000D 010E 00000000 0003 000A8AD8 LD G  IDLE      GDAT MCH14003
000D 010F 00000000 0004 000A8FD8 LD G  IDLE      GDAT MCH14004
000D 0110 00000000 0005 000A94D8 LD G  IDLE      GDAT MCH14005
```

In the previous example, all SLUs for the MCH named MCH1 are listed including those that are idle (CID=0).

The following display is produced when **DLU [LUNM=]MCH15001 FMT2** is entered.

```
     IFN|
MPID RPID CID      VCN  ADDRESS  LUOPT LUST LUTO TYPE SLUNAME  PLUNAME
000D 010C 00000000 0001 000A80D8 LD NP IDLE       PAD MCH15001 A06TSO
```

In the previous example, only the LU named MCH1P001 is listed.

General notes for DLU:

1) The **IFN|RPID** column identifies either the interface number for the physical MCH link on an IBM XTP router (REMOTE TYPE=XTP,IFNUM=xx,...) or the PCE ID of the TCP/IP component for a Cisco XOT router (REMOTE TYPE=XOT,...) to which the SLU is connected.

2) The **SESSINIT/CID** column identifies either how the LU/VC session was established, that is, by an INBOUND or OUTBOUND call (SHOWSESSSINIT) or the VTAM connection identifier when the SLU is bound.

3) The **VCN** column identifies the HNAS Virtual Circuit Number. For PVCs only, the first digit of the VCN will be replaced by the character 'P'.

For XTP, the VCN is the index into the MCH Address Vector Table (AVT) for the VC. The AVT is used to locate the VC control block when a packet is sent or received across the TCPIP socket connection.

For XTP PVCs, the index is the same as the Circuit Identifier carried in the PVC Info packet. The same Circuit Identifier will be used in all subsequent packets sent and received.

For XTP Callout SVCs, the index is the same as the Circuit Identifier carried in the Outbound Call Request packet created by HNAS. The index is created from the first available slot in the AVT starting at the end and working backwards.

For XTP Callin SVCs, the index is assigned based on an available slot in the AVT starting at the beginning and working forward. The index is not related to the Circuit Identifier received from the router.

For XOT, this is an internal value used for information only.

For XOT PVCs, the VCN is the same as the LCN supplied in the PVC= operand which will be the LCN carried in the PVC Setup packet and all subsequent packets sent and received.

For XOT Callout SVCs, the VCN is always set to 0001 which is then used as the LCN carried in the Outbound Call Request packet created by HNAS and all subsequent packets sent and received.

For XOT Callin SVCs, the VCN is the LCN that is received from the router in the Inbound Call Request packet. It will be used as the LCN in all subsequent packets sent and received.

4) The ***xxx*NAME/ADDRESS** column identifies either a TYPE=XOT|XTP REMOTE definition statement name (SHOWNAME or SHOWNAME=RTR), a TYPE=MCH|XTP REMOTE definition statement (SHOWNAME=MCH), a TYPE=SPU REMOTE definition statement (SHOWNAME=SPU) or the LUB address (SHOWADDR). A RTRNAME|MCHNAME is displayed only if the SLU has an active VC connection. An SPUNAME is displayed only if the SLU is defined in the LUNAME= operand of a TYPE=SPU REMOTE definition statement.

5) The **LUOPT** column identifies HNAS options that are currently active for the SLU. A blank indicates that the option is not active. The LUOPT column also displays SLU connect status information. The following options can be displayed.

```
LDOGP
|||||
|||||<- P => LU is permanently assigned
||||    * => LU is dynamically assigned
||||
||||<-- G => GATE=GENERAL
|||     N => GATE=NO
|||
|||<--- O => ACB is open
||<---- D => TRCLU DBK
|<----- L => TRCLU ON
```

6) The **LUST** column identifies the current SLU state. The following states can be displayed.

BN	PLU BIND issued	**BND**	SLU is bound
CLR	SLU is bound, PLU CLEAR issued	**DTA**	SLU data traffic active
IDLE	SLU is inactive but online	**OFLN**	SLU is offline from VARY OFF
RQS	SLU is unbound, REQSESS issued	**SD**	SLU is bound, PLU SHUTD issued
SDC	SLU is bound, SLU SHUTC issued	**SDT**	SLU is bound, PLU SDT issued
UNB	SLU is bound, PLU UNBIND issued	**UBNS**	SLU is bound, SLU UNBIND issued

7) The **LUTO** column identifies the active timer if one is running. The following timer values can be displayed.

DLAY	QLLC DACTLU/ACTLU delay, ACTLU issued on timer expiration	**REXP**	QLLC PIU sent to SLU, response expected
RQSD	QLLC REQSHUT sent to VTAM, SHUTDOWN expected	**RQSE**	QLLC REQSESS sent to VTAM, BIND expected

Note: The LUTO column was introduced into 240 with Enhancement APAR 2400078.

8) The **TYPE** column identifies the SLU type. The following types can be displayed.

GDAT	GATE data SLU	**GATE**	GATE control SLU
IDLE	SLU LLC not resolved	**PCNE**	Type 1 SLU session
PAD	Transparent SLU		

DMAP Command - Display Module Map

(changed for V2R1M0)
(changed for V2R2M0)
(changed for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DMAP	**[{*module*|** **ALL|TRACE|APAR}]**	**C** **C**	**220** **230**

This command will display a HNAS module map for all or selected modules, trace and APAR assignment information. The REFID field of the DMAP report heading identifies This command is also used to identify modules with traces t locate

You may enter an HNAS ***module*** name to restrict the display to a single module.

- Enter **DMAP TRACE** to display the trace entry identifies for all HNAS modules.
- Enter **DMAP APAR** to display the APAR entry identifiers for all HNAS modules.

  HNAS automatically executes the **DMAP APAR** command at initialization time *with no delays* so that the display of system maintenance that is logged in the HNAS SYSPRINT can be viewed using an SDSF panel.

  Additionally, during the initialization pass, the DMAP APAR command creates a table that is sorted in APAR ID order so that it can be displayed using the **DNAS APAR** command. The **DMAP APAR** command displays APARs in module name order rather than APAR ID order.

  Note: The **DMAP APAR** command that is executed unconditionally at HNAS startup no longer logs output in SYSPRINT. This was done to reduce SYSPRINT log activity during startup. If you wish to see DMAP APAR output, you can issue this command once the console input prompt message has been displayed.

Note: If an APAR number is unused (skipped), is special or is deferred to a later release, **NASEND** will be displayed as the module name.

Note: The **DMAP APAR** and **DNAS APAR** commands will display HNASMACX and/or HNASOBJX as an APARID when CustomUserMods are included in the HNAS product distribution. The HNAS distribution process has been modified to generate an 'APARID' of HNASMACX and/or HNASOBJX if a custom MACLIB and/or OBJLIB has been included in the distribution. These APARIDs are contained in the **NASEND** module and will be displayed in the HNAS SYSPRINT when the DMAP APAR console command is executed when HNAS is started. These APARIDs will also display whenever the DNAS APAR console command is executed. These APARIDs are intended to provide an additional reminder that custom modifications are on the HNAS system and, hence, caution should be exercised when applying maintenance. Please refer to the DNAS console command description for additional informa-

tion. Note also that the majority of customers don't have custom enhancements (Custom-UserMods).

- Enter **DMAP ALL** (which is treated as **DMAP TRACE APAR**) to display the trace and APAR entry identifiers for all HNAS modules.

  If the **DMAP** command is entered without any parameters, a list of all modules that comprise the HNAS system will be displayed. When all modules are displayed, they are listed alphabetically, left to right, top to bottom.

  Refer to the example heading information and samples on the following page for additional information:

The following is an example of the heading and information produced by the **DMAP** command for a single module.

```
MODULE     ADDRESS  SIZE  REFNO  REFOF  REFID  LGTM
|          |        |     |      |      |      |
|          |        |     |      |      |      |<- time logged in trace
|          |        |     |      |      |         entry (YES or NO)
|          |        |     |      |      |
|          |        |     |      |      |<- macro identifier can be
|          |        |     |      |         TRACE entry ID,
|          |        |     |      |         APAR entry ID or optional
|          |        |     |      |         CUSTOM MOD entry ID
|          |        |     |      |
|          |        |     |      |<- macro call offset in module
|          |        |     |
|          |        |     |<- macro call reference number
|          |        |
|          |        |<- module size
|          |
|          |<- module address
|
|<- module name
```

The following display is produced when **DMAP NASTCP** is entered.

```
MODULE     ADDRESS  SIZE  REFNO  REFOF  REFID     LGTM
NASTCP   000260D8  6542     01   58B6  TRIDIPRM   YES
                         02   597A  TRIDSBFR    NO
                         03   5B02  TRIDTDAT    NO
                         04   5C32  TRIDIBFR   YES
                         05   5D58  TRIDRBFR    NO
                         06   5F10  TRIDTDAT    NO
                         01   6078  2200003
```

The following display is produced when **DMAP APAR** is entered.

```
MODULE     ADDRESS  SIZE  REFNO  REFOF  REFID     LGTM
CONSDMAP 00016FA0  1600     01   1030  2300004
CONSDNAS 000193A0  0690     01   02E0  2300004
MCHINI   00048500  16A0     01   1096  2300004
NASCNFG  0006F6F0  7400     01   268C  2300001
NASCONS  000110C0  27B0     01   0126  2300004
NASMAIN  00007230  9E90     01   003C  2300000
NASTCP   0003BE80  7490     01   13E6  2300002
NASUTIL  00035CA0  5790     01   0860  2300004
XOTUT1   00068760  0E80     01   0BEA  2300003
```

The following display is produced when **DMAP** is entered without a module name.

MODULE	ADDRESS	MODULE	ADDRESS	MODULE	ADDRESS
CNFGACRT	0009A130	CNFGAFLT	0009A300	CNFGALMT	0009A730
CNFGAPNM	0009AC20	CNFGBFLM	0009B0F0	CNFGBFSZ	0009B3B0
CNFGCNCM	0009B5C0	CNFGCNCT	0009B7C0	CNFGCNLM	0009BB00
CNFGCNPR	0009BDB0	CNFGCNPW	0009BFA0	CNFGCTCP	0009C1A0
CNFGCUD	0009C910	CNFGCUD0	0009CE20	CNFGDCAD	0009D300
CNFGDLNM	0009DD60	CNFGDTAD	0009D850	CNFGDXNM	0009E4A0
CNFGFAC	0009EBC0	CNFGGATE	0009F140	CNFGHOME	0009F410
CNFGIDBK	000A01B0	CNFGIDNM	000A0470	CNFGIDTO	0009FC30
CNFGIFNM	000A07E0	CNFGINIT	000A0B70	CNFGIPAD	000A1300
CNFGLGTB	000A1ED0	CNFGLLC0	000A2410	CNFGLLC2	000A2780
CNFGLLC3	000A2AE0	CNFGLLC4	000A2EE0	CNFGLLC5	000A3250
CNFGLULM	000A3640	CNFGLUNM	000A3AC0	CNFGMBIT	000A5020
CNFGMSLM	000A52C0	CNFGMXDT	000A5730	CNFGNSNM	000A5A00
CNFGOPTS	000A5C00	CNFGPACE	000A9240	CNFGPAD	000A9650
CNFGPADP	000A9910	CNFGPKSZ	000A9D00	CNFGPORT	000AA050
CNFGPRLM	000AA710	CNFGPROT	000AA920	CNFGPRSW	000AAED0
CNFGPVC	000AB3C0	CNFGPWPR	000ACB80	CNFGRTIN	000ACF10
CNFGRTOT	000AD5B0	CNFGSCLM	000ADD40	CNFGSUBA	000ADFA0
CNFGSUBD	000AE330	CNFGSVC0	000AE720	CNFGSVC3	000AF7C0
CNFGSVC4	000B0590	CNFGSVC5	000B0F30	CNFGSYSL	000B2000
CNFGTAP	000B2A50	CNFGTPNM	000B2F40	CNFGTRAN	000B3270
CNFGTRLM	000B3670	CNFGTRTR	000B38C0	CNFGTYPE	000B4230
CNFGUSTB	000B49C0	CNFGVCLM	000B4F50	CNFGVTAM	000B5770
CONSALRM	00015FE0	CONSCID	00017580	CONSCLD	00017910
CONSCLG	00017C00	CONSDAD	00017EF0	CONSDLCL	00018560
CONSDLP	000193E0	CONSDLU	000195D0	CONSDMAP	0001AA10
CONSDMCH	0001C180	CONSDMEM	0001CD00	CONSDNAS	0001D340
CONSDNWD	0001E0C0	CONSDPCE	0001F000	CONSDPRM	00020010
CONSDRMT	00020CB0	CONSDSTS	00023B80	CONSDTRC	00024220
CONSDVC	00024760	CONSEXEC	00025910	CONSFIND	00026A40
CONSHELP	00027410	CONSIFN	00027DF0	CONSIPAD	00028180
CONSLNAM	00028510	CONSLNCT	000287E0	CONSLUN	00028A90
CONSLUNM	00028E10	CONSMLCL	000290E0	CONSMMEM	0002AC00
CONSMON	0002B2C0	CONSMRMT	0002CB00	CONSPAUS	00035470
CONSPID	00035860	CONSPING	00035BF0	CONSPRNT	000373C0
CONSPWTO	00037DD0	CONSQUIT	00038100	CONSRCON	00038E60
CONSRNAM	00039060	CONSSHOW	00039330	CONSSMSG	00039610
CONSSTAT	00039960	CONSTAD	0003A260	CONSTALL	0003A6F0
CONSTBFR	0003AFF0	CONSTBRQ	0003B900	CONSTBST	0003BC10
CONSTCFG	0003BF20	CONSTCON	0003C280	CONSTDAT	0003C800
CONSTDBG	0003D100	CONSTDBK	0003D400	CONSTDSP	0003D730
CONSTIO	0003E070	CONSTLU	0003E960	CONSTLUQ	0003F730
CONSTMCH	000401D0	CONSTMCX	00040D40	CONSTPCE	000418A0
CONSTPRT	00042160	CONSTRTR	00042670	CONSTSUB	00046520
CONSTTSK	00046830	CONSTVC	00046B50	CONSTVCQ	00047580
CONSTWTO	00047F60	CONSUSER	00048260	CONSVARY	00048730
CONSVCN	00049F20	MCHBFR	0005DE40	MCHFCI	0005F2E0

```
MCHHDTRQ 0005F720   MCHHGTRQ 0005F9C0   MCHHIDRQ 00060070
MCHHL0RQ 00060240   MCHHL2RQ 00060B20   MCHHL3RQ 00060EB0
MCHHL4RQ 00061830   MCHHL5RQ 00061F70   MCHHRQ   00062810
MCHHRSP  00063480   MCHINI   000636F0   MCHLUIN  00065150
MCHNRQB  00065C10   MCHNRQC  00066020   MCHPVCI  00066BB0
MCHRL3RR 000672B0   MCHSOL   00067DA0   MCHSTRT  00069760
MCHSUP   00069DB0   MCHSVCI  0006A290   MCHTMR   0006A850
MCHTR    0006B4C0   MCHTRXD  0006C0F0   MCHTTBLS 0006C2E0
MCHUT1   0006D570   MCHUT2   0006E050   NASCNFG  000917D0
NASCONS  00012F50   NASEND   000B5AB0   NASMAIN  00007000
NASNETV  0004A290   NASPRNT  000534D0   NASTCP   000543F0
NASUTIL  0004CC70   OSUT1    0006EDE0   QLSSCP   0006F250
VCCLAC   00073830   VCCLEAR  00073A00   VCCLRQ   000744D0
VCDAT    00075890   VCDATQ   000762B0   VCINT    00076940
VCRESET  00076D70   VCTR     000774A0   VCUT1    00078120
VTMEXIT  00078D60   VTMRCV1  00079E60   VTMSND1  0007B1D0
VTMSND2  0007C620   VTMTR    0007D870   VTMUT1   0007F890
XFCNFGWA 000925C8   XFNASWA  0000C0B8   XOTBXM   00080F50
XOTBXM2  00081F00   XOTDTDC  00082780   XOTFCDC  00082D90
XOTGTCC  00083840   XOTGTDC  000840A0   XOTINI   00084720
XOTRCV   00084870   XOTSTP   00086000   XOTTR    000862A0
XOTUT1   00088220   XOTXMTC  0008B450   XTPBXM   0008C130
XTPDTDC  0008CC10   XTPFCDC  0008D110   XTPGTCC  0008DAB0
XTPGTDC  0008E1F0   XTPINI   0008E820   XTPRCV   0008E970
XTPSTP   0008F760   XTPTR    0008FA20   XTPUT1   00090B60
XTPXMTC  00090EA0
```

DMCH Command - Display Multi-Channel Link

(changed for V2R2M0)
(changed for V2R3M0)

for a specific MCH

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*mchname*]	DMCH	[*mchname*]	C	230
[LUN=*minlun*[-*maxlun*]]		[{SHOWNAME|SHOWADDR}]	N	230
		[{FMT1|FMT2|FMT3}]	A	230

or for a collection of MCHs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	DMCH	[{SHOWNAME|SHOWADDR}]	N	230
[IFN=*minifn*[-*maxifn*]]		[{FMT1|FMT2|FTM3}]	A	230
[LUN=*minlun*[-*maxlun*]]		[ALLID]		
		[ALLIFN]		
		[ALLLUN]		

This command will display HNAS MCH state information.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting DMCH processing to the named MCH only for the specified **LUN=** values.

- When **LUN=*minlun*[-*maxlun*]** is specified, DMCH processing is restricted to the selected MCHXs on the selected MCHs. If **LUN=0** is specified, DMCH processing is for all MCHXs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), DMCH processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, DMCH processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, DMCH processing is restricted to the selected PCEs. If **ID=0** is specified, DMCH processing is for all PCEs.

- The **SHOWNAME** operand (the default) requests that the name of the TYPE=MCH|XTP REMOTE definition statement is to be displayed in the NAME/ADDRESS column.

- The **SHOWADDR** operand requests that the memory address of the MCH control block is to be displayed in the NAME/ADDRESS column.

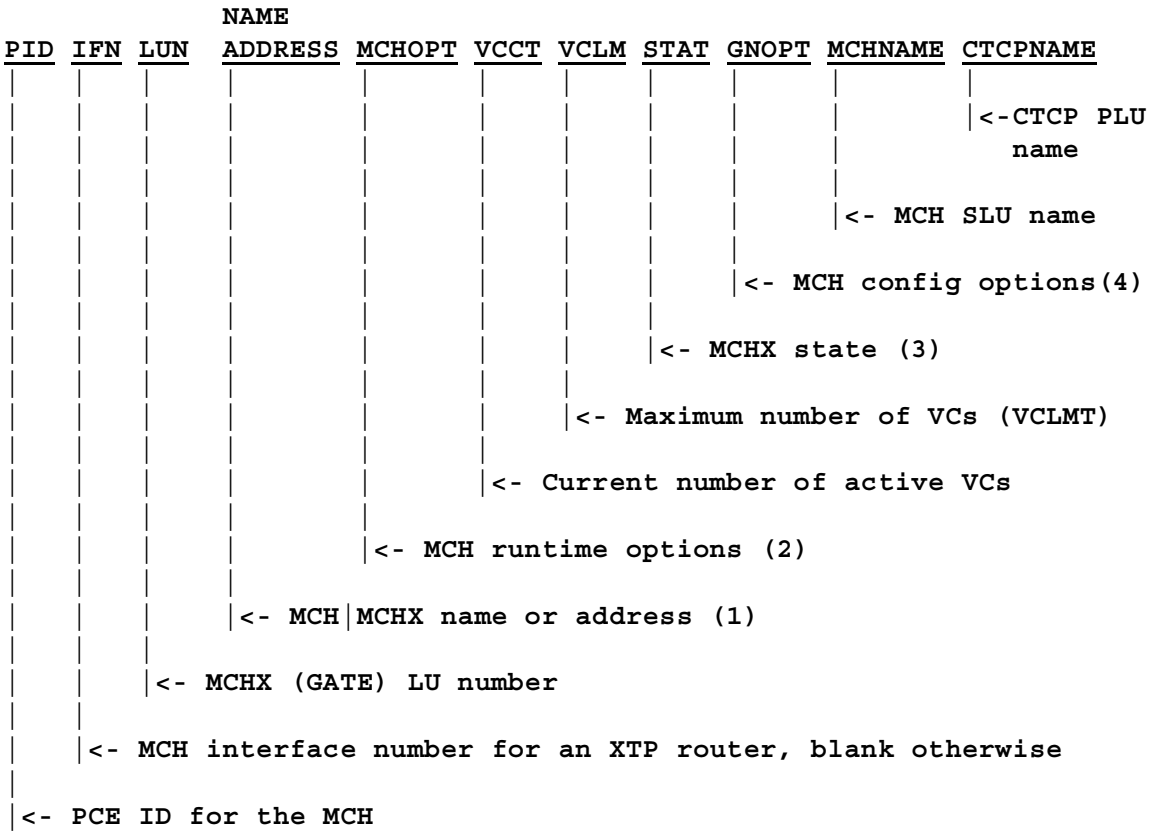
- The **FMT1**, **FMT2** and **FMT3** operands (for format 1, 2 and 3) control the type of information that is displayed. Currently, **FMT1** (the default) is the same as **SHOWNAME** and **FMT2** is the

same as **SHOWADDR**. **FMT3**, which is valid for the LOCAL console only, causes the DLU command to be executed after the DMCH command completes. This will provide a display of the selected MCHs and all LUs that are associated with those MCHs. If FMT3 is entered from a REMOTE console, the DMCH command will be rejected.

Note: The SHOW*xxxx* and FMT*x* operands are processed in the order specified. The last value you specify is the winner. For example, if you enter SHOWADDR FMT1, SHOWNAME will override SHOWADDR because FMT1 includes SHOWNAME.

- The **ALLID** parameter is treated as though RNM= omitted and ID=0 were in effect even if they are not. This means that all PCEs are scanned for their MCH components.
- The **ALLIFN** parameter is treated as though IFN=0 was in effect even if it is not. This means that all MCHs are displayed for each PCE.
- The **ALLLUN** parameter is treated as though LUN=0 was in effect even if it is not. This means that all MCHXs are displayed for each MCH.

The following is an example of the heading and information produced by the **DMCH** command.



The following display is produced when **DMCH BPM** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
PID  IFN LUN NAME     MCHOPT  VCCT VCLM STAT GNOPT MCHNAME  CTCPNAME
0008 003     R1M3     M IOFIO 0005 0256 ACT  G I
          01          MD                O          NASC00A3
0008 004     R1M4     M IOFIO 0000 0064 ACT  G I
          01          MD                O          NASC00A4
000D         MCH1     M IOFIO 0005 0128 ACT  G I
          01          MD                O          NASC0006
```

The following display is produced when **DMCH BPM FMT2** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
PID  IFN LUN ADDRESS  MCHOPT  VCCT VCLM STAT GNOPT MCHNAME  CTCPNAME
0008 003     00096200 M IOFIO 0005 0256 ACT  G I
          01 0009CCC0 MD                O          NASC00A3
0008 004     0009EA18 M IOFIO 0000 0064 ACT  G I
          01 000A0E60 MD                O          NASC00A4
000D         000A12B8 M IOFIO 0005 0128 ACT  G I
          01 000A7B78 MD                O          NASC0006
```

The following display is produced when **DMCH ID=8 IFN=3 LUN=1** is entered.

```
PID  IFN LUN NAME     MCHOPT  VCCT VCLM STAT GNOPT MCHNAME  CTCPNAME
0008 003     R1M3     M IOFIO 0005 0256 ACT  G I
          01          MD                O          NASC00A3
```

General notes for DMCH:

1) The **NAME/ADDRESS** column identifies either the MCH name (SHOWNAME) or the MCH address (SHOWADDR). The name that is used comes from the TYPE=MCH|XTP REMOTE definition statement.

2) The **MCHOPT** column identifies HNAS options that are currently active for the MCH or MCHX. A blank indicates that the option is not active. The following options can be displayed for the MCH.

```
M.IOFIO
|||||||
|||||||<- O => TRCMCH OCLR
||||||<-- I => TRCMCH ICLR
|||||<--- F => TRCMCH ICRF
||||<---- O => TRCMCH OCR
|||<----- I => TRCMCH ICR
||<------ reserved
|<------- M => TRCMCH ON
```

The following options can be displayed for the MCHX.

```
MD.....
|||||||
|||||||<- reserved
||||||<-- reserved
|||||<--- reserved
||||<---- reserved
|||<----- reserved
||<------ D => TRCMCHX DBK
|<------- M => TRCMCHX ON
```

3) The STAT column identifies the current MCH and MCHX state.

For an MCH, the state can be **ACT** or **OFLN**. ACT is displayed if the MCH was configured active via the INIT=ACT operand (the default) or made active via the VARY ON console command. OFLN is displayed if the MCH was configured inactive via the INIT=IDLE operand or made inactive via the VARY OFF console command.

For an MCHX, a blank indicates state inactivity. The following states can be displayed.

```
OENC..
||||||
||||||<- reserved
|||||<-- reserved
||||<--- C => GATEFC closed
|||<---- N => GATEFC nogo
||<----- E => Permanent open error
|<------ O => ACB is open
```

4) The **GNOPT** column identifies the MCH (REMOTE) configuration options. A blank indicates that the option is not active. The following options can be displayed.

```
GSPC.
|||||
|||||<- reserved
||||<-- Y => CONNECT=YES
|||      C => CONNECT=CUD0
|||      S => CONNECT=SUBD
|||<--- I => PAD=INTEG
||       X => PAD=TRANSP
||<---- S => SUBADDR=YES
|<----- G => GATE=GENERAL
```

DNAS Command - Display HNAS Information

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DNAS	**[{APAR [*aparid*]|** **USERMODS [*probid*]}** **{JCL|DDNAMES}** **{NOIDENT}**	**C** **A** **A** **A**	**220** **230** **240** **240**

This command will display information about the HNAS system you are using. Information provided includes the HNAS distribution level, type, creation date and time as well as maintenance base and recent APAR ID assignment and APAR ID history.

- Enter **DNAS** to display the HNAS product distribution system and maintenance information summary. The following is an example of the display produced when **DNAS** is entered with no arguments for a permanent distribution (effective with APAR 2400103).

```
HOST NAS INFORMATION FOLLOWS
     HNAS VERSION=V2R4M0 DIST=NON-SMP                                  1
     HNAS PROGRAM RUNNING UNDER z/OS 01.10.00                          2
     HNAS PRODUCT INSTALLED UNDER z/OS 01.10.00                        3
     HNAS PRODUCT CREATED UNDER z/OS 01.11.00                          4
     DNAS COMMAND ENTERED AT 18:54:01 ON 2010/12/01                    5
     HNAS PROGRAM STARTED AT 18:54:01 ON 2010/12/01                    6
     HNAS PRODUCT INSTALLED AT 08:12:00 ON 2010/11/29                  7
     HNAS PRODUCT CREATED AT 08:19:12 ON 2010/11/28                    8
     HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400106        9
     MOST RECENT MAINTENANCE APPLIED IS APAR 2400106                  10
     AUTH=000  SHIPID=1100000011199999                                11
     CUSTID=SFD_99999                                                 12
     CUSTINFO=COMM-PRO ASSOCIATES                                     13
     MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31                   14
     DATAFONO SUPPORT IS INCLUDED                                     15
     CUSTMAC=COMM1.TEST.HNASMAC                                       16
     CUSTOBJ=COMM1.TEST.HNASOBJ                                       17
                                                                      18
                                                                      19
     APARID   MAINTENANCE STATUS                                      20
     ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400106              21

     - or -

     APARID   MAINTENANCE STATUS                                      20
     aparid   NOT INSTALLED                                           21
     :
     aparid   NOT INSTALLED                                            n
```

For a trial distribution, DNAS display records 11 and 14 will appear as follows (for example):

```
AUTH=032D SHIPID=1100000011199999                              11
TRIAL PERIOD EXPIRATION DATE IS 2010/12/31                     14
```

Note: Line numbers to the right of the DNAS display above are for reference only.

Display line 1 shows the HNAS product Version, Release and Modification level and the distribution media type (SMP/E or NON-SMP).

Display line 2 shows the host system and version level under which HNAS is running.

Display line 3 shows the host system and version level under which HNAS was installed. This may be different than display line 2 if the HNAS load library is moved from one system to another.

Display line 4 shows the host system and version level under which HNAS was created. Effective with APAR 2400103, display line 4 should show z/OS 01.11.00.

Display line 5 shows the time and date when the DNAS command was entered.

Display line 6 shows the time and date when the HNAS program was started.

Display line 7 shows the time and date when the HNAS product was installed.

Display line 8 shows the time and date when the HNAS distribution was created.

Display line 9 shows the highest APAR number that was included in the HNAS distribution when it was created (***shipped*_APARID**).

Display line 10 shows the highest APAR number of maintenance applied to HNAS after the distribution was installed at the customer site (***applied*_APARID**). The *shipped*_APARID and the *applied*_APARID will be the same if no maintenance has been applied since the original HNAS install. If there are no missing APARs between the *shipped*_APARID and the *applied*_APARID (all APAR numbers are consecutive) then line 21 will be displayed as '**ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400104**'. If, however, there are 'holes' between the *shipped*_APARID and the *applied*_APARID then the missing APAR(s) are displayed staring at line 21 as '***aparid*** **NOT INSTALLED**'.

Note: Display line 10 represents the highest apar number on the system (not the most recent APAR applied). If this number is different than the number on line 9, the customer support representative will know that maintenance was applied to the product (HNAS customer support requires that the DNAS output be provided whenever a problem is reported or a product refresh or upgrade is requested). While the default DNAS command does not identify all of the APARS that are on the HNAS system (unless DNAS APAR is specified), it does identify the base APAR level (***shipped*_APARID**) and highest

numbered APAR applied (***applied*_APARID**). To identify all APARs on the HNAS system you would have to execute the **DNAS APAR** or **DMAP APAR** console command.

Display line 11 shows the customer SHIPID and number of months (M), weeks (W) or days (D) that the product is authorized to use starting from the distribution creation date shown in display line 8. AUTH=000 implies there is no authorization expiration (the distribution is permanent).

Display line 12 is displayed if optional customer identification was provided when the HNAS distribution was created.

Display line 13 is displayed if optional customer information was provided when the HNAS distribution was created.

Display line 14 shows the HNAS Maintenance/Use Anniversary Date (EOMDATE) for a permanent distribution (normally one year after the program was purchased or renewed) or the Trial Period Expiration Date (EOTDATE) for trial distribution.

Note: The Maintenance/Use Anniversary Date is based upon the Prepaid Maintenance or Purchase Order period that was in effect when the edistribution was generated. The date (*yyyy*/*mm*/*dd*) is normally updated as HNAS product refreshes are deployed with current billing information.

Display line 15 is displayed if HNAS was distributed with optional Datafono support.

Display lines 16 and/or 17 are displayed if a custom MACLIB and/or OBJLIB were used to create the HNAS distribution.

Note: For a permanent distribution, when an EOMKEY= value is supplied via the PARM= operand on the HNAS EXEC statement or as an argument for the MMEM console command, display lines 11, 14 and 18 will be modified as follows:

```
AUTH=000  SHIPID=1100000011199999 EMKYID=1100000011199999     11
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31*               14
EOMKEY=1760932717613627 IS IN EFFECT                          18
```

The EMKYID= value on display line 11 is the SHIPID= value extracted from the EOMKEY= value. This value *can be* different from the SHIPID= value on display line 11 in the first 2 digits only. These digits are reserved for Comm-Pro use.

The date displayed on display line 14 is the extended EOMDATE which is flagged with an asterisk (*).

The EOMKEY= text on display line 18 shows the EOMKEY= that was provided via the EXEC PARM= operand or the MMEM console command.

Note: EOMKEY= parameter logic was introduced into 240 via APAR 2400095.

Note: For a trial distribution, when an EOTKEY= value is supplied via the PARM= operand on the HNAS EXEC statement or as an argument for the MMEM console command, display lines 11, 14 and 18 will be modified as follows:

```
AUTH=060D SHIPID=1100000011199999 ETKYID=1100000011199999     11
TRIAL PERIOD EXPIRATION DATE IS 2011/01/28*                   14
EOTKEY=4961000737880526 IS IN EFFECT                          18
```

The ETKYID= value on display line 11 is the SHIPID= value extracted from the EOTKEY= value. This value *can be* different from the SHIPID= value on display line 11 in the first 2 digits only. These digits are reserved for Comm-Pro use.

The date displayed on display line 14 is the extended EOTDATE which is flagged with an asterisk (*).

The EOTKEY= text on display line 18 shows the EOTKEY= that was provided via the EXEC PARM= operand or the MMEM console command.

Note: EOTKEY= parameter logic was introduced into 240 via APAR 2400106.

General notes for DNAS:

1) The AUTH= field in display line 11 identifies whether or not the distribution has a trial period. A value of 000 indicates that the distribution is permanent (no expiration) while non-zero values identify the number of Days, Weeks or Months that the distribution will operate beyond the "HNAS PRODUCT CREATED AT *hh*:*mm*:*ss* ON *yyyy*/*mm*/*dd*" date on display line 8.

```
AUTH=limit{M|W|D}    (limit=months|weeks|days, 0=>PERMANENT >0=>TRIAL)
```

The SHIPID= field in display line 11 contains a Comm-Pro assigned customer reference number as well as flags depicting distribution types (SMP/E versus non-SMP), the presence of custom user modifications (enhancements) within the distribution and how the EOMDATE will be processed.

```
SHIPID=distribution-type_CustomUserMod-flags_custid#
0000000011099999 <-- sample non-smpe with custom macro/object.
||     |||||     |
||     |||||     |<--- CUSTOMER ID REFERENCE NUMBER: nnnnn
||     |||||
||     |||||<-------- DATAFONO FLAG
||     ||||            1 => DATAFONO SUPPORT INCLUDED
||     ||||            0 => NO DATAFONO SUPPORT INCLUDED
||     ||||
||     ||||<---------- CUSTOBJ FLAG
||     |||             1 => CUSTOM OBJECT CHANGES INCLUDED
||     |||             0 => NO CUSTOM OBJECT CHANGES INCLUDED
||     |||
||     |||<---------- CUSTMAC FLAG
||     ||              1 => CUSTOM MACRO CHANGES INCLUDED
||     ||              0 => NO CUSTOM MACRO CHANGES INCLUDED
||     ||
||     ||<----------- CUSTSRC FLAG
||     |               1 => CUSTOM SOURCE CHANGES INCLUDED
||     |               0 => NO CUSTOM SOURCE CHANGES INCLUDED
||     |
||     |<------------ SMPDIST FLAG
||                     1 => SMP/E DISTRIBUTION
||                     0 => NON-SMP DISTRIBUTION
||
||<----------------- CP FLAGS
                      0|1 => INTERNAL USE.
```

2) Following display line 20, DNAS will now show a list of missing APAR IDs if the *applied*_APARID is different than the *shipped*_APARID that was incorporated when the HNAS distribution was created. For example, if you were shipped HNAS with APAR 2400059 incorporated and you then installed APAR 2400062 separately, the following would be displayed if you did not also install the 'pre-reqs' for APAR 2400062:

```
APARID   MAINTENANCE STATUS
2400060  NOT INSTALLED
```

```
2400061  NOT INSTALLED
```

3) If there is no missing maintenance, that is, if there are no gaps between the *shipped*_APARID value and the *applied*_APARID value (all pre-reqs were also installed), line 21 will display as follows:

```
ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400104
```

4) The missing APARID logic is only executed when DNAS is entered *without* the NOIDENT parameter.

5) Enter **DNAS APAR** to display all APAR entry identifiers (like 2400095) and custom modification entry identifiers (like P04057A) for all HNAS modules in alpha-numeric ID order in addition to the standard DNAS identification summary.

6) The **DMAP APAR** command creates a table at HNAS initialization that is sorted in alpha-numeric APARID order so that it can be displayed using the **DNAS APAR** command. Note that this enhancement was introduced by APAR 2300004.

The following is an example of the display produced when **DNAS APAR** is entered.

```
HOST NAS INFORMATION FOLLOWS
     HNAS VERSION=V2R4M0 DIST=NON-SMP                                 1
     HNAS PROGRAM RUNNING UNDER z/OS 01.10.00                         2
     HNAS PRODUCT INSTALLED UNDER z/OS 01.10.00                       3
     HNAS PRODUCT CREATED UNDER z/OS 01.11.00                         4
     DNAS COMMAND ENTERED AT 07:33:27 ON 2010/09/16                   5
     HNAS PROGRAM STARTED AT 07:33:26 ON 2010/09/16                   6
     HNAS PRODUCT INSTALLED AT 07:29:00 ON 2010/09/16                 7
     HNAS PRODUCT CREATED AT 07:08:18 ON 2010/09/16                   8
     HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400104       9
     MOST RECENT MAINTENANCE APPLIED IS APAR 2400104                 10
     AUTH=000  SHIPID=1100000011199999                               11
     CUSTID=SFD_99999                                                12
     CUSTINFO=COMM-PRO ASSOCIATES                                    13
     MAINTENANCE/USE ANNIVERSARY DATE IS 2010/09/31                  14
     DATAFONO SUPPORT IS INCLUDED                                    15
     CUSTMAC=COMM1.TEST.HNASMAC                                      16
     CUSTOBJ=COMM1.TEST.HNASOBJ                                      17
                                                                     18
                                                                     19
     APARID   MAINTENANCE STATUS                                     20
     ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400104             21
                                                                     22
     APARID   MODULE   APPLIED MAINTENANCE                           23
     2400000  NASMAIN
     2400001  CNFGOPTS
              CNFGPRLM
              CNFGPRSW
              CONSDMAP
              CONSDMCH
              CONSDNAS
```

```
              CONSPRNT
              CONSTALL
              CONSTPRT
              CONSTRTR
              CONSVARY
              NASCNFG
              NASCONS
              NASMAIN
              NASPRNT
              NASUTIL
       :
       2400104  CONSDNAS
```

7) If an APAR number is unused (skipped), is special or is deferred to a later release, **NASEND** will be displayed as the module name.

8) The **DNAS APAR** and **DMAP APAR** commands now display HNASMACX and/or HNASOBJX as an APARID when CustomUserMods are included in the HNAS product distribution. The HNAS distribution process has been modified to generate an 'APARID' of HNASMACX and/or HNASOBJX if a custom MACLIB and/or OBJLIB has been included in the distribution. These APARIDs are contained in the **NASEND** module and will be displayed in the HNAS SYSPRINT when the DMAP APAR console command is executed when HNAS is started. These APARIDs will also display whenever the DNAS APAR console command is executed. These APARIDs are intended to provide an additional reminder that custom modifications are on the HNAS system and, hence, caution should be exercised when applying maintenance. Please refer to the DMAP console command description for additional information. Note also that the majority of customers don't have custom enhancements (CustomUserMods).

9) Enter **DNAS USERMODS** to display only custom modification entry identifiers or problem identifiers (for example, P04057A) for all HNAS modules in alpha-numeric ID order as well as the standard summary. Note that this enhancement was introduced by APAR 2300065.

The following is an example of the display produced when **`DNAS USERMODS`** is entered.

```
HOST NAS INFORMATION FOLLOWS
     HNAS VERSION=V2R4M0 DIST=NON-SMP                                 1
     HNAS PROGRAM RUNNING UNDER z/OS 01.10.00                         2
     HNAS PRODUCT INSTALLED UNDER z/OS 01.10.00                       3
     HNAS PRODUCT CREATED UNDER z/OS 01.11.00                         4
     DNAS COMMAND ENTERED AT 07:33:27 ON 2010/09/16                   5
     HNAS PROGRAM STARTED AT 07:33:26 ON 2010/09/16                   6
     HNAS PRODUCT INSTALLED AT 07:29:00 ON 2010/09/16                 7
     HNAS PRODUCT CREATED AT 07:08:18 ON 2010/09/16                   8
     HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400104       9
     MOST RECENT MAINTENANCE APPLIED IS APAR 2400104                 10
     AUTH=000  SHIPID=1100000011199999                               11
     CUSTID=SFD_99999                                                12
     CUSTINFO=COMM-PRO ASSOCIATES                                    13
     MAINTENANCE/USE ANNIVERSARY DATE IS 2010/09/31                  14
```

```
DATAFONO SUPPORT IS INCLUDED                                    15
CUSTMAC=COMM1.TEST.HNASMAC                                      16
CUSTOBJ=COMM1.TEST.HNASOBJ                                      17
                                                                18
                                                                19
APARID   MAINTENANCE STATUS                                     20
ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400104             21
                                                                22
APARID   MODULE   APPLIED MAINTENANCE                           23
NO USERMODS PRESENT                                             24

- or -

 APARID   MODULE   APPLIED MAINTENANCE                          23
 probid   module                                                 24
:
 probid   module                                                 n
```

10) Enter **DNAS APAR *aparid*** or **USERMODS *probid*** to restrict the display of applied maintenance to the specified ***aparid*** or ***probid*** *only.*

For example: **DNAS APAR 2400104** will produce following display:

```
HOST NAS INFORMATION FOLLOWS
    HNAS VERSION=V2R4M0 DIST=NON-SMP                             1
    HNAS PROGRAM RUNNING UNDER z/OS 01.10.00                     2
    HNAS PRODUCT INSTALLED UNDER z/OS 01.10.00                   3
    HNAS PRODUCT CREATED UNDER z/OS 01.11.00                     4
    DNAS COMMAND ENTERED AT 07:33:27 ON 2010/09/16               5
    HNAS PROGRAM STARTED AT 07:33:26 ON 2010/09/16               6
    HNAS PRODUCT INSTALLED AT 07:29:00 ON 2010/09/16             7
    HNAS PRODUCT CREATED AT 07:08:18 ON 2010/09/16               8
    HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400104   9
    MOST RECENT MAINTENANCE APPLIED IS APAR 2400104             10
    AUTH=000  SHIPID=1100000011199999                           11
    CUSTID=SFD_99999                                            12
    CUSTINFO=COMM-PRO ASSOCIATES                                13
    MAINTENANCE/USE ANNIVERSARY DATE IS 2010/09/31              14
    DATAFONO SUPPORT IS INCLUDED                                15
    CUSTMAC=COMM1.TEST.HNASMAC                                  16
    CUSTOBJ=COMM1.TEST.HNASOBJ                                  17
                                                                18
                                                                19
    APARID   MAINTENANCE STATUS                                 20
    ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400104         21
                                                                22
    APARID   MODULE   APPLIED MAINTENANCE                       23
    2400104  CONSDNAS
```

11) Enter **DNAS NOIDENT *parms*** to inhibit display lines 1 through 19.

12) Enter **DNAS JCL|DDNAMES** to display the HNAS jobname and all DDNAMEs in use by HNAS.

For example: **DNAS JCL** will produce following display:

```
HOST NAS INFORMATION FOLLOWS
LDNAS240 DDNAME   VOLSER UNIT   TTR-STC  STB-UCB
         STEPLIB  USER04 AA4    8E1C9000 80F3F8C8
                  USER04 AA4    8DF6D800 80F3F8C8
                  USER04 AA4    8DF85800 80F3F8C8
         VTAMLIB  Z4RES1 A80    8DF9D800 80F3E5A8
                  USER04 AA4    8DFB5800 80F3F8C8
                  USER04 AA4    8DFCD800 80F3F8C8
         SYSPRINT SPOOL  JES    8DFE5800 80000000
         SYSPRNT1 SPOOL  JES    8DFFD800 80000000
         SYSPRNT2 WORK08 AB8    8E015800 90F40368
         SYSPRNT3 SPOOL  JES    8E02D800 80000000
         SYSABEND SPOOL  JES    8E045800 80000000
         SYSUDUMP SPOOL  JES    8E05D800 80000000
         AUTH     USER04 AA4    8E075800 80F3F8C8
         NEWDEFN  USER04 AA4    8E08D800 80F3F8C8
         CMDLIST1 USER04 AA4    8E0E1800 80F3F8C8
         CMDLIST2 USER04 AA4    8D431800 80F3F8C8
         CMDLIST3 USER04 AA4    8D449800 80F3F8C8
         CMDLIST4 USER04 AA4    8D461800 80F3F8C8
         CONFIG   SPOOL  JES    8D479800 80000000
 ALL DDNAMES LISTED
```

Note: The **DNAS** command (without an argument) is now executed when HNAS is started without having to be specified in the CONCMDQ= operand. This is done unconditionally regardless of the commands listed in CONCMDQ=. As a result, DNAS is no longer the default when CONCMDQ= is omitted. The DNAS command is also executed uncondionally during FASTRUN execution without having to specify CONCMDQ as a FASTRUN follower in the PARM= operand. FASTRUN or FASTRUN CONCMDQ are treated the same way effective with APAR 2400103.

DNWDF Command - Display New Definition CDF (Privileged)

(added for V2R3M0)
(changed for V2R4M0)

for the BUILD definition statement

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DNWDF	**BUILD**	**A**	**230**
		[ALLPARMS] |		
		[NOSHOWPFX]		
		[ALRMCART[=]]		
		[ALRMFLTR[=]]		
		[ALRMLMTS[=]]		
		[APPLNAME[=]]		
		[BFRLMT[=]]		
		[BFRSIZ[=]]		
		[CONCMDQ[=]]		
		[CONLMT[=]]		
		[CONPRMT[=]]		
		[CONPSWD[=]]		
		[IDLETO[=]]		
		[LOGTAB[=]]		
		[MSGLMT[=]]		
		[NASNAME[=]]		
		[OPTIONS[=]]		
		[PRTLMT[=]]		
		[PRTSWLST[=]]	**N**	**240**
		[PULSE[=]]	**A**	**240**
		[SCHEDULE[=]]	**A**	**240**
		[TCPNAME[=]]		
		[TRCLMT[=]]		
		[TRCTRAP[=]]		
		[USSTAB[=]]		

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[VCLMT[=]]		

or for a specific LOCAL definition statement

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LNM=*lclname*	**DNWDF**	**[*lclname*]**	**A**	**230**
		LOCAL		
		[ALLPARMS] |		
		[NOSHOWPFX]		
		[INIT[=]]		
		[IPADDR[=]]		
		[OPTIONS[=]]		
		[PORT[=]]		
		[RTEIN[=]]		
		[RTEOUT[=]]		
		[SOCLMT[=]]		
		[TCPNAME[=]]		
		[TYPE[=]]		

or for a specific REMOTE definition statement

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
RNM=*rmtname*	**DNWDF**	**[*rmtname*]**	**A**	**230**
		REMOTE		
		[ALLPARMS] |		
		[NOSHOWPFX]		
		[APPLNAME[=]]		
		[CONNECT[=]]		
		[CTCP[=]]		
		[CUD[=]]		
		[CUD0[=]]		

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[DCEADDR[=]]		
		[DFLNAME[=]]	**N**	**240**
		[DFXNAME[=]]	**N**	**240**
		[DTEADDR[=]]		
		[FAC[=]]		
		[GATE[=]]		
		[HOME[=]]		
		[IDBLK[=]]		
		[IDNUM[=]]		
		[IDLETO[=]]		
		[IFNUM[=]]		
		[INIT[=]]		
		[IPADDR[=]]		
		[LLC0[=]]		
		[LLC3[=]]		
		[LLC4[=]]		
		[LLC5[=]]		
		[LOGTAB[=]]		
		[LUNAME[=]]		
		[MAXDATA[=]]		
		[MBITCHN[=]]		
		[OPTIONS[=]]		
		[PACE[=]]		
		[PAD[=]]		
		[PADPARM[=]]		
		[PORT[=]]		
		[PROTOCOL[=]]	**N**	**240**
		[PVC[=]]		
		[PWPROT[=]]		

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[SUBADDR[=]]		
		[SUBD[=]]		
		[SVC0[=]]		
		[SVC3[=]]		
		[SVC4[=]]		
		[SVC5[=]]		
		[SYSL[=]]		
		[TAP[=]]		
		[TRAN[=]]		
		[TYPE[=]]		
		[USSTAB[=]]		
		[VCLMT[=]]		

or for the entire CDF

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DNWDF	**[ALL]**	**A**	**230**
		[SHOWCOMMENTS]		
		[NOSHOWPFX]		

This command will display the records that were specified in the CDF or modified later via the MLCL or MRMT command.

To summarize, you may enter a list of CDF definition statement operands that you want to display as follows:

DNWDF Action Matrix

PARAMETER	ACTION	BUILD	LOCAL	REMOTE
ALL or omitted	**Display value for all CDF operand records.**			
SHOWCOMMENTS	**Display CDF comment records when ALL CDF operand records are displayed.**			
NOSHOWPFX	**Inhibit NEWDEFN prefix control information from being displayed when CDF records are displayed.**			
ALRMCART=	**Display value for ALRMCART= operand.**		**I**	**I**

PARAMETER	ACTION	BUILD	LOCAL	REMOTE
ALRMFLTR=	Display value for ALRMFLTR= operand.		I	I
ALRMLMTS=	Display value for ALRMLMTS= operand.		I	I
APPLNAME=	Display value for APPLNAME= operand.		I	
BFRLMT=	Display value for BFRLMT= operand.		I	I
BFRSIZ=	Display value for BFRSIZ= operand.		I	I
CONCMDQ=	Display value for CONCMDQ= operand.		I	I
CONLMT=	Display value for CONLMT= operand.		I	I
CONNECT=	Display value for CONNECT= operand.	I	I	
CONPRMT=	Display value for CONPRMT= operand.		I	I
CONPSWD=	Display value for CONPSWD= operand.		I	I
CTCP=	Display value for CTCP= operand.	I	I	
CUD=	Display value for CUD= operand.	I	I	
CUD0=	Display value for CUD0= operand.	I	I	
DCEADRR=	Display value for DCEADDR= operand.	I	I	
DFLNAME=	Display value for DFLNAME= operand.	I	I	
DFXNAME=	Display value for DFXNAME= operand.	I	I	
DTEADDR=	Display value for DTEADDR= operand.	I	I	
FAC=	Display value for FAC= operand.	I	I	
GATE=	Display value for GATE= operand.	I	I	
HOME=	Display value for HOME= operand.	I	I	
IDBLK=	Display value for IDBLK= operand.	I	I	
IDNUM=	Display value for IDNUM= operand.	I	I	
IDLETO=	Display value for IDLETO= operand.		I	
IFNUM=	Display value for IFNUM= operand.	I	I	
INIT=	Display value for INIT= operand.	I		
IPADDR=	Display value for IPADDR= operand.	I		
LLC0=	Display value for LLC0= operand.	I	I	
LLC3=	Display value for LLC3= operand.	I	I	
LLC4=	Display value for LLC4= operand.	I	I	
LLC5=	Display value for LLC5= operand.	I	I	
LOGTAB=	Display value for LOGTAB= operand.		I	
LUNAME=	Display value for LUNAME= operand.	I	I	
MAXDATA=	Display value for MAXDATA= operand.	I	I	

PARAMETER	ACTION	BUILD	LOCAL	REMOTE
MBITCHN=	Display value for MBITCHN= operand.	I	I	
MSGLMT=	Display value for MSGLMT= operand.		I	I
NASNAME=	Display value for NASNAME= operand.		I	I
OPTIONS=	Display value for OPTIONS= operand.			
PACE=	Display value for PACE= operand.	I	I	
PAD=	Display value for PAD= operand.	I	I	
PADPARM=	Display value for PADPARM= operand.	I	I	
PKTSIZ=	Display value for PKTSIZ= operand.	I	I	
PORT=	Display value for PORT= operand.	I		
PROTOCOL=	Display value for PROTOCOL= operand.	I	I	
PRTLMT=	Display value for PRTLMT= operand.		I	I
PRTSWLST=	Display value for PRTSWLST= operand.		I	I
PULSE=	Display value for PULSE= operand.		I	I
PVC=	Display value for PVC= operand.	I	I	
PWPROT=	Display value for PWPROT= operand.	I	I	
RTEIN=	Display value for RTEIN= operand.	I		I
RTEOUT=	Display value for RTEOUT= operand.	I		I
SCHEDULE=	Display value for SCHEDULE= operand.		I	I
SOCLMT=	Display value for SOCLMT= operand.	I		I
SUBADDR=	Display value for SUBADDR= operand.	I	I	
SUBD=	Display value for SUBD= operand.	I	I	
SVC0=	Display value for SVC0= operand.	I	I	
SVC3=	Display value for SVC3= operand.	I	I	
SVC4=	Display value for SVC4= operand.	I	I	
SVC5=	Display value for SVC5= operand.	I	I	
SYSL=	Display value for SYSL= operand.	I	I	
TAP=	Display value for TAP= operand.	I	I	
TCPNAME=	Display value for TCPNAME= operand.			I
TRAN=	Display value for TRAN= operand.	I	I	
TRCLMT=	Display value for TRCLMT= operand.		I	I
TRCTRAP=	Display value for TRCTRAP= operand.		I	I
TYPE=	Display value for TYPE= operand.	I		
USSTAB=	Display value for USSTAB= operand.		I	

PARAMETER	ACTION	BUILD	LOCAL	REMOTE
VCLMT=	**Display value for VCLMT= operand.**		**I**	

Legend: For the definition statement *type* column: I=> Invalid, blank=> supported (see Console Command Parameter Action Matrix Keys description on page CONS-21 for additional information).

If no operand list is specified, **ALL** is assumed which causes all CDF records to be displayed.

The following is an example of the heading and information produced by the **DRMT** command.

```
<NEWDEFN  PREFIX>
RECADDR   DDOOSSEE NAME      DEF       OPER
|         |        |         |         |
|         |        |         |         |
|         |        |         |         |<- CDF Operand
|         |        |         |
|         |        |         |<- CDF definition statement type
|         |        |             (BUILD, LOCAL, REMOTE)
|         |        |
|         |        |<- Resource name
|         |
|         |<- DD = definition ID (80=>BUILD, 40=>LOCAL, 20=>REMOTE)
|            OO = operand ID (00=>ALRMCART, 01=>ALRMFLTR, etc.)
|            SS = start of operand data offset
|            EE = end of operand data offset
|
|<- NEWDEFN record memory address
```

The following display is produced when **DNWDF BUILD** is entered. Note that comments are truncated because the NEWDEFN prefix (RECADDR DDOOSSEE) is present.

```
 RECADDR DDOOSSEE NAME     DEF   OPER
0011BA70 8001181F          BUILD ALRMFLTR=(ALLOW,
0011BAD4 80010F1B                NAS3****(P))
0011BB9C 80041619                BFRLMT=500               ; BUFFER POOL REQUIREM
0011BC00 80051619                BFRSIZ=236               ; 236, 708, 280 1078(1
0011C628 80071617                CONLMT=2                 ; TOTAL NUMBER OF RCON
0011C68C 80091722                CONPRMT='ZDTFCS1> '      ; CONSOLE PROMPT.
0011C6F0 800A171B                CONPSWD=XXXX             ;
0011C754 8017161A                IDLETO=1440              ; NNNN MINUTES - INACT
0011C7B8 8024161A                MSGLMT=1000              ; MAX TCP/IP MSG REQUE
0011C81C 8026171F                NASNAME=NAST0TSO         ; OUR NAME
0011C880 802E161D                PRTLMT=2250000           ; TOTAL NUMBER PRINT R
0011CAD8 802F181E                PRTSWLST=(LOOP,
0011CB3C 802F0F1B                SWITCHAT6PM,
0011CBA0 802F0F18                SYSPRNT1,
0011CC04 802F0F18                SYSPRNT2,
0011CC68 802F0F18                SYSPRNT3)
0011CCCC 803E171C                TCPNAME=TCPIP            ; TCP/IP ADDRESS SPACE
0011CD30 8040161D                TRCLMT=0004096           ; TRACE TABLE REQUIREM
0011D0B4 8043161E                USSTAB=ISTINCDT          ; IBM STANDARD USSTAB
```

The following display is produced when **DNWDF BUILD NOSHOWPFX** is entered. Note that comments are not truncated because the NEWDEFN prefix (RECADDR DDOOSSEE) is not present.

```
NAME      DEF   OPER
          BUILD ALRMFLTR=(ALLOW,
                NAS3****(P))
                BFRLMT=500               ; BUFFER POOL REQUIREMENT.
                BFRSIZ=236               ; 236, 708, 280 1078(1024+52)
                CONLMT=2                 ; TOTAL NUMBER OF RCONS
                CONPRMT='ZDTFCS1> '      ; CONSOLE PROMPT.
                CONPSWD=XXXX             ;
                IDLETO=1440              ; NNNN MINUTES - INACTIVITY T/O.
                MSGLMT=1000              ; MAX TCP/IP MSG REQUEST LMT.
                NASNAME=NAST0TSO         ; OUR NAME
                PRTLMT=2250000           ; TOTAL NUMBER PRINT RECORDS
                PRTSWLST=(LOOP,
                SWITCHAT6PM,
                SYSPRNT1,
                SYSPRNT2,
                SYSPRNT3)
                TCPNAME=TCPIP            ; TCP/IP ADDRESS SPACE NAME
                TRCLMT=0004096           ; TRACE TABLE REQUIREMENT
                USSTAB=ISTINCDT          ; IBM STANDARD USSTAB
```

The following display is produced when **DNWDF [LNM=]LXOT LOCAL** is entered.

```
 RECADDR DDOOSSEE NAME     DEF   OPER
0011D3D4 40421417 LXOT     LOCAL TYPE=XOT                 ; OUR TYPE
0011D438 401A1623                IPADDR=10.117.56.171     ; OUR IP ADDRESS
0011D500 402C1418                PORT=1998                ; OUR PORT NUMBER
0011D564 40321527                RTEIN=(MCHCONS/20370009,
0011D5C8 40320F1E                CLEAR/20360002,
0011D62C 40320F1D                SKIP/2036**03,
0011D690 40320F14                MCH1)
0011D758 4033163C                RTEOUT=(R2CNOT1/666612345678901/'RMT DTE #1',
0011D7BC 40330F17                R2CNOT1)
0011D820 40341619                SOCLMT=512               ; 2000 IS DEFAULT.
0011D884 403E171C                TCPNAME=TCPIP            ; TCPIP IS DEFAULT.
0011D94C 4019141C                INIT=(ACTIVE,
0011D9B0 40190F1B                DELAYTIME=1)
```

The following display is produced when **DNWDF [LNM=]LXOT LOCAL NOSHOWPFX** is entered.

```
NAME     DEF   OPER
LXOT     LOCAL TYPE=XOT                 ; OUR TYPE
               IPADDR=10.117.56.171     ; OUR IP ADDRESS
               PORT=1998                ; OUR PORT NUMBER
               RTEIN=(MCHCONS/20370009,
               CLEAR/20360002,
               SKIP/2036**03,
               MCH1)
               RTEOUT=(R2CNOT1/666612345678901/'RMT DTE #1',
               R2CNOT1)
               SOCLMT=512               ; 2000 IS DEFAULT.
               TCPNAME=TCPIP            ; TCPIP IS DEFAULT.
               INIT=(ACTIVE,
               DELAYTIME=1)
```

The following display is produced when **DNWDF [RNM=]MCH1 REMOTE** is entered.

```
 RECADDR DDOOSSEE NAME     DEF    OPER
00888D34 2043151A MCH1     REMOTE TYPE=(MCH)              ; REMOTE TYPE
00888D98 20271722                 OPTIONS=(MCHTMR=60,     ; ELIMINATE DEFAULTS
00888EC4 20270F17                 ECHOFAC)                ; TEST NEW OPTION
00889054 20451519                 VCLMT=(32)              ; ROUTER VC COUNT
0088911C 2029131A                 PAD=(INTEG)             ; INTEGRATED PAD SUPPO
008891E4 202A171C                 PADPARM=(1/1,           ; ESC TO CMD OK.
00889248 202A0F13                 2/1,                    ; ECHO_DEVICE_INPUT -
008892AC 202A0F13                 3/2,                    ; DATA_FORWARD_CHAR =
00889310 202A0F13                 4/0,                    ; DISABLE- FWD=5 SEC I
00889374 202A0F13                 5/2,                    ; DEVICE_FLOW_CTL = XO
008893D8 202A0F14                 7/21,                   ; BREAK = INT, DISCARD
0088943C 202A0F14                 12/1,                   ; PAD____FLOW_CTL = XO
008894A0 202A0F14                 13/4,                   ; AUTO_LF_INS =  WAS 7
00889504 202A0F14                 21/0,                   ; PARITY_TREAT = NO CH
00889568 202A0F15                 110/0)                  ; DISABLE PAD-8 INACT
08895CC 2032161D                  PWPROT=YESWOCC          ;
00889694 2013141D                 GATE=(GENERAL)          ; GATE SUPPORT
008896F8 2014141A                 HOME=(LXOT)             ; THIS DEFAULT IS LXOT
0088975C 2008171B                 CONNECT=(NO)            ; NO FAST CONNECT
008897C0 2023171C                 MBITCHN=(YES)           ; EACH RU IS --- SOMET
00889824 200B1418                 CTCP=(85,               ; CUD0 TO LLC/CTCP MAP
00889888 200B0F12                 80)
008898EC 200D1418                 CUD0=(01,               ; CUD0 VALUES.
00889950 200D0F14                 NULL)
008899B4 2036171C                 SUBADDR=(YES)           ;    PP DIGIT PROCESSI
00889A18 201C141A                 LLC0=(NONE)
00889A7C 201F1417                 LLC4=(4)                ; P DIGIT 4 SETS LLC 4
00889AE0 20201418                 LLC5=NONE
00889B44 2040141A                 TRAN=(EVEN)
00889C0C 20221620                 LUNAME=(NASC0006,       ; CALLOUT RESOURCES
00889C70 20220F18                 XXXXXXCS,
00889CD4 20220F18                 XXXXXXDS)
00889D9C 20031820                 APPLNAME=(A06TSO,       ; TSO=0,CONSOLE=1,SOL.
00889E00 20030F17                 CONSOLE,
00889E64 20030F16                 MCHSOL)
00889EC8 2044161D                 USSTAB=ZZUSST1          ; CP CUSTOM USSTAB
0088A24C 200C1317                 CUD=NONE                ; NO CALL USER DATA.
0088A2B0 200E171B                 DCEADDR=NONE            ; NO CALLING DCEADDR.
0088A314 20121317                 FAC=NONE                ; NO FACILITIES.
0088A4A4 20311319                 PVC=(NONE)              ; NO PVC SUPPORT.
0088A634 203A1419                 SVC3=ALLOW
0088A6FC 20381418                 SVC0=(10,               ; CALLING ADDRESS
0088A8F0 20380F18                 MCH10001,
0088A954 20380F19                 MCH10002/,
0088A9B8 20380F19                 MCH10003/,
0088AA1C 20380F2D                 MCH10004/66666666666666I01//,
0088AA80 20380F2E                 MCH10005/666612345678901O/MXT1)
0088AC10 203B1417                 SVC4=(2,                ; GATE DIAL IN.
0088AC74 203B0F18                 MCH14001,               ;
0088ACD8 203B0F18                 MCH14002)
0088B188 203C1417                 SVC5=(5,                ; CALLING ADDRESS
0088B1EC 203C0F18                 MCH15001,
0088B250 203C0F18                 MCH15002,
0088B2B4 203C0F18                 MCH15003,
0088B318 203C0F18                 MCH15004,
```

```
0088B37C 203C0F18               MCH15005)
0088B3E0 203D141F               SYSL=(SUBD=01/0,         ; CALLED ADDRESS
0088B444 203D0F19               SUBD=02/0,
0088B4A8 203D0F19               SUBD=03/0,
0088B50C 203D0F19               SUBD=04/0,
0088B570 203D0F19               SUBD=05/2,
0088B5D4 203D0F19               SUBD=06/0,
0088B638 203D0F19               SUBD=09/1,               ; PP DIGITS=09=CONSOLE
0088B69C 203D0F16               NULL/0)
```

DPARM Command - Display Command Modifiers and Parameters

(changed for V2R2M0)
(changed for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DPARM	[ALL] |	N	114
		[CID[=]]		
		[CLDADDR[=]]		
		[CLGADDR[=]]		
		[EXEC]	N	220
		[ID[=]]		
		[IFN[=]]		
		[IPADDR[=]]		
		[LNCT[=]]		
		[LNM[=]]	N	220
		[LUN[=]]		
		[LUNM[=]]	N	220
		[MODIFIERS]	A	230
		[RNM[=]]	C	220
		[VCN[=]]		

This command will display the console command modifiers and/or EXEC start parameters that are currently in effect. Note that the value of command modifiers remain in effect until they are changed or until the console session is terminated.

To summarize, you may enter a DPARM parameter that requests a specific action to be performed as follows:

DPARM Action Matrix

PARAMETER	ACTION
ALL or omitted	Display value for all command modifiers and all start parameters.
CID=	Display value for CID= modifier.
CLDADDR=	Display value for CLDADDR= modifier.
CLGADDR=	Display value for CLGADDR= modifier.
EXEC	Display value for EXEC start parameters only (withhold MODIFIERS).

PARAMETER	ACTION
ID=	Display value for ID= modifier.
IFN=	Display value for IFN= modifier.
IPADDR=	Display value for IPADDR= modifier.
LNCT=	Display value for LNCT= modifier.
LNM=	Display value for LNM= modifier.
LUN=	Display value for LUN= modifier.
LUNM=	Display value for LUNM= modifier.
MODIFIERS	Display value for command modifiers only (withhold EXEC).
RNM=	Display value for RNM= modifier.
VCN=	Display value for VCN= modifier.

If no command modifier list is specified, **ALL** is assumed. Note that entering **DPARM [ALL]** is equivalent to entering **DPARM CID= CLDADDR= CLGADDR= ... VCN=**.

The following is an example of the heading and information produced by the **DPARM** command.

```
COMMAND MODIFIERS FOLLOW        VALID VALUES
CID=00000001-00000002           0 OR 00000001-FFFFFFFE
CLDADDR=123                     1-15 DECIMAL DIGITS
CLGADDR=21                      1-15 DECIMAL DIGITS
ID=0001-0002                    0 OR 0001-0026
IFN=001-002                     0 OR 001-255
IPADDR=001.001.001.001(01001)   STD SOCKET FORMAT
LNM=LXOT                        1-8 CHARACTER SYMBOL
LNCT=00023                      0 OR 00001-65535
LUN=001-002                     0 OR 01-28
LUNM=MCH15001                   1-8 CHARACTER SYMBOL
RNM=MCH1                        1-8 CHARACTER SYMBOL
VCN=00001-00002                 0 OR 0001-4095

EXEC PARAMETERS FOLLOW
APFXEQ ON
APFMEMSP=(230
         229)
DUMP OFF
FASTRUN OFF
GENNWDF ON
MONTAP OFF
PFXWTO Z230SFD6 OFF
PRNTCNFG ON
PRNTDATE OFF
```

```
PRNTLU ON
PRNTON
PRNTQLLC ON
PRNTSYS ON
PRNTTCP ON
PRNTVC ON
PRNTVTAM ON
PRNTXOT ON
PRNTXTP ON
RMTCONS PRIV
SHOWCNFG ON
SHOWCONS OFF
SHOWERR
SHOWMORE OFF
STATS OFF
STATS NOTYPES
SVRSTRT OFF
TRCBFR ALLOFF           <- see Note2 below.
TRCBFR NOTYPES
TRCBFRQ OFF
TRCBST OFF
TRCCNFG OFF
TRCCONS ALLOFF
TRCDATA ALLOFF          <- see Note2 below.
TRCDATA NOTYPES
TRCDBK OFF
TRCDISP ALLOFF          <- see Note2 below.
TRCDISP NOTYPES
TRCDISP NQDQ ALLOFF
TRCIO ALLOFF            <- see Note2 below.
TRCIO NOTYPES
TRCLU ALLON             <- see Note2 below.
TRCLU MINDATA
TRCMCH ALLON            <- see Note2 below.
TRCMCH ICRF
TRCMCHX ALLON           <- see Note2 below.
TRCPRNT OFF
TRCSUBR OFF
TRCSUBR NOEVENTS
TRCTASK OFF
TRCVC ALLON             <- see Note2 below.
TRCVC MINDATA
TRCWTO OFF
USEMDFY ON USEWTOR OFF
USENETV OFF             <- parameter reserved for future use
```

Note1: For those console commands that can toggle the value established by a start parameter (*global* action), the ALLON|ALLOFF argument may be used instead of the ON|OFF argument. This is only the case when the same command can control both a *global* and *local* action. For these commands, ALLON|ALLOFF is used to manipulate the *global* action while

ON|OFF is used to manipulate the *local* action. For example, the TRCLU ALLON and TRCLU ALLOFF console commands turn on and off, respectively, the value set by the TRCLU start parameter.

Note2: The DPARM EXEC command will display ALLON or ALLOFF as a start parameter follower if that can be manipulated by a console command using the ALLON or ALLOFF argument. For a start parameter that can be manipulated by a console command using the ON or OFF argument, ON or OFF will continue to be displayed by DPARM EXEC.

The following are console commands that can manipulate start parameters with the ALLON|ALLOFF argument: TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCMCH, TRCMCHX and TRCVC.

The following sample display is produced when **`DPARM CLDADDR= IPADDR=`** is entered as the last command of the following command sequence:

```
CLDADDR=1024888
IPADDR=5.6.7.8(1098)
DPARM CLDADDR= IPADDR=

COMMAND MODIFIERS FOLLOW        VALID VALUES
CLDADDR=1024888                 1-15 DECIMAL DIGITS
IPADDR=005.006.007.008(01098)   STANDARD SOCKET FORMAT
```

DPCE Command - Display Process Control Element

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DPCE	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[{SHOWNAME|SHOWADDR}]	**N**	**230**
		[{FMT1|FMT2}]	**N**	**114**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**DPCE**	**[*rmtname*]**	**C**	**230**
		[{SHOWNAME|SHOWADDR}]	**N**	**230**
		[{FMT1|FMT2}]	**N**	**114**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**DPCE**	**[{SHOWNAME|SHOWADDR}]**	**N**	**230**
		[{FMT1|FMT2}]	**N**	**114**
		[ALLID]	**N**	**114**
		[PROT={XOT|XTP}]		

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[STATE={ACTIVE| **CLOSING|** **GCSCLR|** **GCSCON|** **GCSERP|** **GCSINI|** **GCSSVR|** **IDLE|** **OFFLINE|** **TCPACC|** **TCPACT|** **TCPBND|** **TCPCLS|** **TCPCON|** **TCPGBN|** **TCPGSN|** **TCPGTC|** **TCPGVS|** **TCPINI|** **TCPIOC|** **TCPLSN|** **TCPSEL|** **TCPSOC|** **TCPSSO|** **TCPTKS]**	**N** **N**	**220** **240**
		[TRACE={OFF|ON}]	**N**	**230**
		[TYPE={CONS|TCP|TMR| **UTIL|XOT|XTP}]**	**N**	**220**

This command will display HNAS PCE state information. If only the HEADER line is displayed after the command is entered it indicates that nothing matched the display criteria as requested by the command parameters entered or those currently in effect.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs DPCE processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled ‘PCE name as a Suffix Command Modifier’ on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value restricting DPCE processing to the named REMOTE only.

- When **ID=*minid*[*-maxid*]** is specified, DPCE processing is restricted to the selected PCEs. If **ID=0** is specified, DPCE processing is for all PCEs.

- The **SHOWNAME** operand (the default) requests that the name of the PCE is to be displayed in the NAME/ADDRESS column. For a server PCE, the name comes from the LOCAL definition statement. For a client PCE, the name comes from the REMOTE definition statement.

For all other PCEs, the name is generated internally.

- The **SHOWADDR** operand requests that the memory address of the PCE is to be displayed in the NAME/ADDRESS column.

- The **FMT1** and **FMT2** operands (for format 1 and 2) control the type of information that is displayed. Currently, **FMT1** (the default) is the same as **SHOWNAME** and **FMT2** is the same as **SHOWADDR**.

  Note: The SHOW*xxxx* and FMT*x* operands are processed in the order specified. The last value you specify is the winner. For example, if you enter SHOWADDR FMT1, SHOWNAME will override SHOWADDR because FMT1 includes SHOWNAME.

- The **ALLID** parameter is treated as though RNM= omitted and ID=0 were in effect even if they are not. This means that all PCEs are displayed subject to filtering by the **PROT=**, **STATE=**, **TRACE=** and/or **TYPE=** operands.

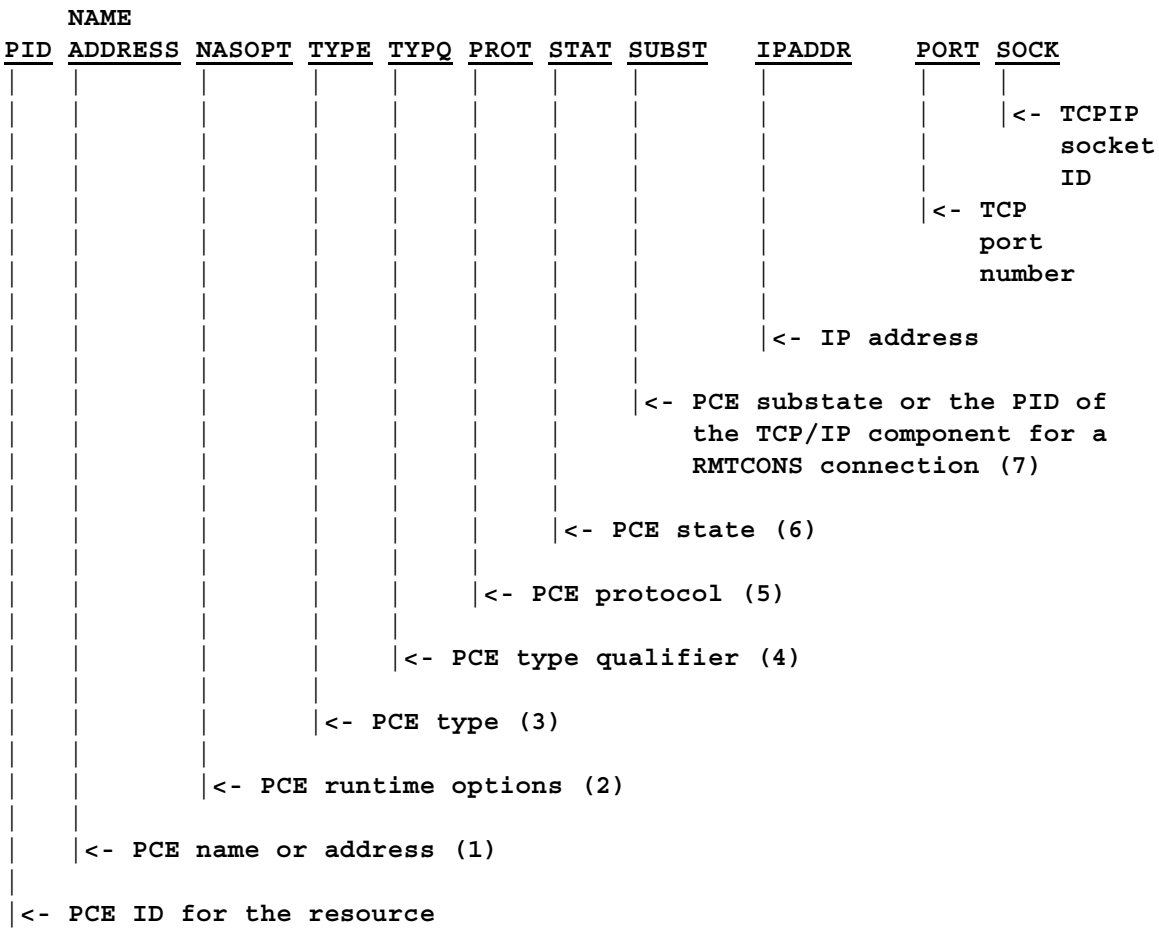
- The **TYPE=** parameter is used to restrict the display to a specific PCE type.

- The **PROT=** parameter is used to restrict the display to a specific protocol for **TYPE=TCP** PCEs.

- The **STATE=** parameter is used to restrict the display to a specific state for **TYPE=TCP** PCEs.

- The **TRACE=** parameter is used to restrict the display to PCEs being traced (**ON**) or not being traced (**OFF**).

The following is an example of the heading and information produced by the **DPCE** command.



The following display is produced when **DPCE BPM** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
PID  NAME      NASOPT TYPE TYPQ PROT STAT SUBST IPADDR          PORT  SOCK
0001 WATMRPCE         TMR
0002 WACONPCE        CONS  LCL       ACT
0003 WANTVPCE        NETV           IDLE
0004 WAPNGPCE         TCP PING  XOT IDLE       255.255.255.255 65535
0006 XTPUTIL          XTP
0007 XOTUTIL          XOT
0008 LXTP             TCP  LCL  XTP TSEL  C    192.040.060.002 03065 0000
0009 LXOT             TCP  LCL  XOT TSEL  C    009.117.056.221 01998 0001
000A R1M3             TCP  RMT  XTP  ACT WC S  192.040.060.001 16476 001E
000B R2CNOT1          TCP  RMT  XOT  ACT TC S  009.117.056.100 01223 0003
000C R2CNOT1          TCP  RMT  XOT  ACT  C S  009.117.056.100 01224 0004
000D R2CNOT1          TCP  RMT  XOT TSOC       009.117.056.100 01998
000E R2CNOT1          TCP  RMT  XOT TSOC       009.117.056.100 01998
000F R2CNOT2          TCP  RMT  XOT OFLN       009.117.056.100 01998
0010 R2CNOT2          TCP  RMT  XOT OFLN       009.117.056.100 01998
0011 R2CNOT2          TCP  RMT  XOT OFLN       009.117.056.100 01998
0012 R2CNOT2          TCP  RMT  XOT OFLN       009.117.056.100 01998
0013 R2CNIN           TCP  RMT  XOT  ACT  C S  009.117.056.100 11002 001F
0014 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0015 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0016 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0017 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0018 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0019 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
001A R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
001B R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
001C R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
001D R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
001E R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
001F R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0020 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0021 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0022 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0023 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0024 R2CNIN           TCP  RMT  XOT IDLE       009.117.056.100 65534
0025 MCH1             TCP  MCH  XOT
0026 RCON0026        CONS  RMT  XOT  ACT  0011 009.117.056.100 11002 001F
0005 WANASPCE        UTIL
```

The following display is produced when **DPCE BPM FMT2** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
PID  ADDRESS  NASOPT TYPE TYPQ PROT STAT SUBST IPADDR          PORT  SOCK
0001 0000C300        TMR
0002 0000C3B8       CONS  LCL       ACT
0003 0000D0E8       NETV           IDLE
0004 0000DaF8        TCP PING  XOT IDLE       255.255.255.255 65535
0006 00069F20        XTP
0007 00069F88        XOT
0008 00069FF0        TCP  LCL  XTP TSEL  C    192.040.060.002 03065 0000
0009 0006A41C        TCP  LCL  XOT TSEL  C    009.117.056.221 01998 0001
000A 0006A848        TCP  RMT  XTP  ACT WC S  192.040.060.001 16476 001E
000B 0006B2E4        TCP  RMT  XOT  ACT TC S  009.117.056.100 01223 0003
000C 0006B978        TCP  RMT  XOT  ACT  C S  009.117.056.100 01224 0004
000D 0006C00C        TCP  RMT  XOT TSOC       009.117.056.100 01998
000E 0006C6A0        TCP  RMT  XOT TSOC       009.117.056.100 01998
000F 0006CD34        TCP  RMT  XOT OFLN       009.117.056.100 01998
0010 0006D3C8        TCP  RMT  XOT OFLN       009.117.056.100 01998
0011 0006DA5C        TCP  RMT  XOT OFLN       009.117.056.100 01998
0012 0006E0F0        TCP  RMT  XOT OFLN       009.117.056.100 01998
0013 0006E784        TCP  RMT  XOT  ACT  C S  009.117.056.100 11002 001F
0014 0006FB40        TCP  RMT  XOT IDLE       009.117.056.100 65534
0015 000701D4        TCP  RMT  XOT IDLE       009.117.056.100 65534
0016 00070868        TCP  RMT  XOT IDLE       009.117.056.100 65534
0017 00070EFC        TCP  RMT  XOT IDLE       009.117.056.100 65534
0018 00071590        TCP  RMT  XOT IDLE       009.117.056.100 65534
0019 00071C24        TCP  RMT  XOT IDLE       009.117.056.100 65534
001A 000722B8        TCP  RMT  XOT IDLE       009.117.056.100 65534
001B 0007294C        TCP  RMT  XOT IDLE       009.117.056.100 65534
001C 00072FE0        TCP  RMT  XOT IDLE       009.117.056.100 65534
001D 00073674        TCP  RMT  XOT IDLE       009.117.056.100 65534
001E 00073D08        TCP  RMT  XOT IDLE       009.117.056.100 65534
001F 0007439C        TCP  RMT  XOT IDLE       009.117.056.100 65534
0020 00074A30        TCP  RMT  XOT IDLE       009.117.056.100 65534
0021 000750C4        TCP  RMT  XOT IDLE       009.117.056.100 65534
0022 00075758        TCP  RMT  XOT IDLE       009.117.056.100 65534
0023 00075DEC        TCP  RMT  XOT IDLE       009.117.056.100 65534
0024 00076480        TCP  RMT  XOT IDLE       009.117.056.100 65534
0025 00076B14        TCP  MCH  XOT
0026 00076BB0       CONS  RMT  XOT  ACT  0011 009.117.056.100 11002 001F
0005 0000E3B0       UTIL
```

The following display is produced when **DPCE ID=9** is entered.

```
PID  NAME     NASOPT TYPE TYPQ PROT STAT SUBST IPADDR          PORT  SOCK
0009 LXOT            TCP  LCL  XOT TSEL  C    009.117.056.221 01998 0001
```

The following display is produced when **DPCE [RNM=]R2CNOT1** is entered.

```
PID  NAME     NASOPT TYPE TYPQ PROT STAT SUBST IPADDR          PORT  SOCK
000B R2CNOT1         TCP  RMT  XOT  ACT TC S  009.117.056.100 01223 0003
000C R2CNOT1         TCP  RMT  XOT  ACT  C S  009.117.056.100 01224 0004
000D R2CNOT1         TCP  RMT  XOT TSOC       009.117.056.100 01998
000E R2CNOT1         TCP  RMT  XOT TSOC       009.117.056.100 01998
```

The following display is produced when **DPCE [RNM=]R2CNOT1 FMT2** is entered.

```
PID  ADDRESS  NASOPT TYPE TYPQ PROT STAT SUBST IPADDR          PORT  SOCK
000B 0006B2E4        TCP  RMT  XOT  ACT TC S  009.117.056.100 01223 0003
000C 0006B978        TCP  RMT  XOT  ACT  C S  009.117.056.100 01224 0004
000D 0006C00C        TCP  RMT  XOT TSOC       009.117.056.100 01998
000E 0006C6A0        TCP  RMT  XOT TSOC       009.117.056.100 01998
```

General notes for DPCE:

1) The **NAME/ADDRESS** column identifies either the PCE name (SHOWNAME) or the PCE address (SHOWADDR). For TYPE=XOT REMOTE PCEs, the same name can appear for different PCE ID values. This is because the name that is used comes from the REMOTE definition statement whose PORT and VCLMT operand values caused the PCEs to be generated.

2) The **NASOPT** column identifies HNAS options that are currently active for the PCE. A blank indicates that the option is not active. The following options can be displayed.

```
TIBDMS
||||||
||||||<- S => STATS ON
|||||<-- M => MONTAP ON
||||<--- D => TRCDATA ON
|||<---- B => TRCBFR ON
||<----- I => TRCIO ON
|<------ T => TRCDISP ON
```

3) The **TYPE** column identifies the PCE type. The following types can be displayed.

CONS	Console process	**CON***	Console process (you)
TCP	TCP/IP process	**TMR**	Timer process
UTIL	Utility process	**XTP**	XTP process
XOT	XOT process		

Note that if the DPCE command is entered from a remote console, an asterisk (*) will replace the 'S' in the 'CONS' text to identify the remote PCE session receiving the display.

4) The **TYPQ** column identifies the PCE type qualifier. The following qualifiers can be displayed.

LCL	Local connection	**RMT**	Remote connection
MCH	Logical MCH	**SPU**	QLLC SPU
DFL	Datafono Leased SLU pool	**DFS**	Datafono Switched SLU pool

5) The **PROT** column identifies the protocol being used for the PCE connection. The following qualifiers can be displayed.

XTP	IBM's X.25 via TCP	**XOT**	Cisco's X.25 over TCP

6) The **STAT** column identifies the current PCE state (request being executed). The following states can be displayed.

ACT	Process is active	**GCLR**	GCS CLEAR
GCON	GCS CONNECT	**GERP**	GCS ERP
GINI	GCS INIT	**GSVR**	GCS SEVER
IDLE	Process is inactive but online	**OFLN**	Process is offline from VARY OFF or INIT=IDLE
TACC	TCP ACCEPT	**TBND**	TCP BIND
TCLS	TCP CLOSE	**TCON**	TCP CONNECT
TGBN	TCP GETBYNAME	**TGSN**	TCP GETSOCKETNAME
TGTC	TCP GETCLIENTID	**TGVS**	TCP GIVESOCKET
TINI	TCP INIT	**TIOC**	TCP IOCTL
TLSN	TCP LISTEN	**TSEL**	TCP SELECT
TSOC	TCP SOCKET	**TSSO**	TCP SETSOCKETOPT
TTKS	TCP TAKESOCKET		

7) The **SUBST** column identifies the PCE substate. The PCE substate qualifies the state. A blank indicates substate inactivity. The following substates can be displayed.

```
TCXRH
|||||
|||||<- H => Executing TCPIP CANCEL
||||<-- R => Executing TCPIP RECEIVE
|||      S => Executing TCPIP SELECT
|||
|||<--- X => Executing TCPIP SEND
||<---- C => Executing TCPIP Control Request
|<----- T => TAPping Process
         W => TAPping Active, Waiting
         D => Dynamic IP address
```

Note that for a remote console PCE, the **SUBST** will contain the PCE ID of the TCP/IP component for the router to which the remote console is connected.

DRMT Command - Display REMOTE Configuration

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**DRMT**	**[*rmtname*]**	**C**	**230**
		[ALL] |	**N**	**114**
		[APPLNAME[=]]		
		[CONNECT[=]]		
		[CTCP[=]]		
		[CUD[=]]		
		[CUD0[=]]		
		[DCEADDR[=]]		
		[DFLNAME[=]]	**N**	**240**
		[DFXNAME[=]]	**N**	**240**
		[DTEADDR[=]]	**N**	**230**
		[FAC[=]]		
		[GATE[=]]		
		[HOME[=]]	**N**	**220**
		[IDBLK[=]]	**N**	**220**
		[IDNUM[=]]	**N**	**220**
		[IDLETO[=]]	**N**	**230**
		[IFNUM[=]]		
		[INIT[=]]		
		[IPADDR[=]]		
		[LLC0[=]]		
		[LLC3[=]]	**N**	**220**
		[LLC4[=]]		
		[LLC5[=]]		
		[LOGTAB[=]]	**A**	**210**

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[LUNAME[=*sluname*]]		
		[MAXDATA[=]]	**N**	**220**
		[MBITCHN[=]]		
		[OPTIONS[=]]	**A** **C** **C** **C**	**210** **220** **230** **240**
		[PACE[=]]		
		[PAD[=]]		
		[PADPARM[=]]		
		[PORT[=]]		
		[PROTOCOL[=]]	**N**	**240**
		[PVC[=*sluname*]]		
		[PWPROT[=]]		
		[SUBADDR[=]]		
		[SUBD[=]]		
		[SVC0[=*sluname*]]		
		[SVC3[=*spuname*]]	**N**	**220**
		[SVC4[=*sluname*]]		
		[SVC5[=*sluname*]]		
		[SYSL[=]]		
		[TAP[=]]		
		[TRAN[=]]		
		[TYPE[=]]		
		[USSTAB[=]]		
		[VCLMT[=]]		

This command will display the operands that were specified in the CDF or modified later via the MRMT command for the REMOTE definition statement identified by the **RNM=** modifier. If no **RNM=** value is set, the list of all REMOTE definition statements is displayed.

To summarize, you may enter a list of REMOTE definition statement operands that you want to display as follows:

DRMT Action Matrix

PARAMETER	ACTION	XTP	XOT	MCH	MXT	SPU	DFL	DFS	DFX	DMY	SVC
ALL or omitted	Display value for all REMOTE operands.										
APPLNAME=	Display value for APPLNAME= operand.		I		I		I	I	I	I	I
CONNECT=	Display value for CONNECT= operand.		I		I	I	I	I	I	I	I
CTCP=	Display value for CTCP= operand.		I		I	I	I	I	I	I	I
CUD=	Display value for CUD= operand.						I	I	I		
CUD0=	Display value for CUD0= operand.		I		I	I	I	I	I	I	I
DCEADRR=	Display value for DCEADDR= operand.						I	I	I		
DFLNAME=	Display value for DFLNAME= operand.	I	I		I	I	I	I	I	I	I
DFXNAME=	Display value for DFXNAME= operand.	I	I		I	I	I	I	I	I	I
DTEADDR=	Display value for DTEADDR= operand.	I		I			I	I	I		
FAC=	Display value for FAC= operand.						I	I	I		
GATE=	Display value for GATE= operand.		I		I	I	I	I	I	I	I
HOME=	Display value for HOME= operand.				I	I	I	I	I	I	I
IDBLK=	Display value for IDBLK= operand.	I	I	I	I		I	I	I	I	I
IDNUM=	Display value for IDNUM= operand.	I	I	I	I		I	I	I	I	I
IDLETO=	Display value for IDLETO= operand.		I				I	I	I	I	I
IFNUM=	Display value for IFNUM= operand.		I	I	I	I	I	I	I	I	I
INIT=	Display value for INIT= operand.				I	I	I	I	I	I	I
IPADDR=	Display value for IPADDR= operand.			I	I	I	I	I	I		I
LLC0=	Display value for LLC0= operand.		I		I	I	I	I	I	I	I
LLC3=	Display value for LLC3= operand.	I	I		I	I	I	I	I	I	I
LLC4=	Display value for LLC4= operand.		I		I	I	I	I	I	I	I
LLC5=	Display value for LLC5= operand.		I		I	I	I	I	I	I	I
LOGTAB=	Display value for LOGTAB= operand.		I				I	I	I	I	I
LUNAME=	Display value for LUNAME= operand.		I		I				I	I	I
MAXDATA=	Display value for MAXDATA= operand.	I	I	I	I		I	I	I	I	I
MBITCHN=	Display value for MBITCHN= operand.		I		I	I	I	I	I	I	I
OPTIONS=	Display value for OPTIONS= operand.						I	I		I	I
PACE=	Display value for PACE= operand.		I	I	I		I	I	I	I	I
PAD=	Display value for PAD= operand.		I		I	I	I	I	I	I	I

PARAMETER	ACTION	XTP	XOT	MCH	MXT	SPU	DFL	DFS	DFX	DMY	SVC
PADPARM=	Display value for PADPARM= operand.		I			I	I	I	I	I	I
PKTSIZ=	Display value for PKTSIZ= operand.	I		I	I	I	I	I	I	I	I
PORT=	Display value for PORT= operand.			I	I	I	I	I	I		I
PROTOCOL=	Display value for PROTOCOL= operand.				I						I
PVC=	Display value for PVC= operand.		I		I	I	I	I	I	I	I
PWPROT=	Display value for PWPROT= operand.		I		I	I	I	I	I	I	I
SUBADDR=	Display value for SUBADDR= operand.		I		I	I	I	I	I	I	I
SUBD=	Display value for SUBD= operand.		I		I	I	I	I	I	I	I
SVC0=	Display value for SVC0= operand.		I		I	I	I	I	I	I	I
SVC3=	Display value for SVC3= operand.	I	I		I	I	I	I	I	I	I
SVC4=	Display value for SVC4= operand.		I		I	I	I	I	I	I	I
SVC5=	Display value for SVC5= operand.		I		I	I	I	I	I	I	I
SYSL=	Display value for SYSL= operand.		I		I		I	I	I	I	I
TAP=	Display value for TAP= operand.			I	I	I	I	I	I	I	I
TRAN=	Display value for TRAN= operand.		I		I	I	I	I	I	I	I
TYPE=	Display value for TYPE= operand.										I
USSTAB=	Display value for USSTAB= operand.		I				I	I	I	I	I
VCLMT=	Display value for VCLMT= operand.				I	I			I	I	I

Legend: For the REMOTE *type* column: I=> Invalid, blank=> supported (see Console Command Parameter Action Matrix Keys description on page CONS-21 for additional information).

If no operand list is specified, **ALL** is assumed. Note that entering **DRMT [ALL]** is equivalent to entering **DRMT APPLNAME= CONNECT= ... VCLMT=**.

Certain keywords may specify an SLU name (e.g., SVC0=*sluname*), in which case only the list entry that corresponds to the named SLU will be displayed (see example below).

Note: For keywords on a TYPE=MXT REMOTE definition statement, NONE|NULL *will be* displayed if NONE|NULL is coded for the operand in the CDF (e.g., CUD=NONE|NULL). If an parameter is omitted in the CDF, NULL or no value (blank) *can be* displayed (e.g., CUD= omitted displays as CUD=NULL (this is also the case for DCEADDR=, DTEADDR= and PADPARM=) while IDLE= omitted (as well as LOGTAB= and USSTAB=) displays as IDLE= no value (blank).

Note: For keywords on a TYPE=SPU|MCH|XTP|XOT REMOTE definition statement, NULL *can be* displayed when NONE|NULL is coded for the operand in the CDF (e.g.,

CUD=NONE|NULL). If an parameter is omitted in the CDF, a default value or no value *can be* displayed (CUD= omitted displays as CUD= 01000000 while DCEADDR= omitted displays as DCEADDR=).

The following is an example of the heading and information produced by the **DRMT** command.

```
RMTNAME  OPERAND  TYPE
|        |        |
|        |        |<- REMOTE type
|        |
|        |<- Operand/Value or
|            REMOTE address
|
|<- REMOTE name
```

The following display is produced when **DRMT RNM=** is entered (no REMOTE name given).

```
RMTNAME  OPERAND  TYPE
MCH1     0007E140 MCH
MCH1P301 0008F078 SPU
MCH13001 000902C0 SPU
MCH13101 00091508 SPU
Q1990100 00092750 SPU
Q2000100 00093998 SPU
Q2010100 00094BE0 SPU
Q2020100 00095E28 SPU
Q2030100 00097070 SPU
Q2040100 000982B8 SPU
Q2050100 00099500 SPU
Q2060100 0009A748 SPU
Q2070100 0009B990 SPU
Q2080100 0009CBD8 SPU
Q2090100 0009DE20 SPU
Q2100100 0009F068 SPU
Q9910100 000A02B0 SPU
MCH2     000A14F8 MCH
MXT0     000B2430 MXT
MXT1     000B2880 MXT
R2CNOT1  000B2CD0 XOT
R2CNOT2  000B2D9C XOT
R2CNIN   000B2E68 XOT
R3CNIN   000B2F34 XOT
```

The following display is produced when **DRMT** [**RNM=**]**Q9910100** is entered.

```
RMTNAME  OPERAND  TYPE
Q9910100 000A02B0 SPU
         APPLNAME=MCHSOL     (000)
                  A06TSO     (001)
         CUD=NULL
         DCEADDR=NULL
         FAC=NULL
         IDBLK=991
         IDNUM=00001
         IDLETO=00010
         INIT=ACTIVE
         LOGTAB=
         LUNAME=********                          (001)
                Q1990102/201/201/01/PVCMXT     (002)
                Q1990103                       (003)
         MAXDATA=00265
         OPTIONS=CLOTINITYP=NONE
                  PEER
         PACE=NONE
              NONE
         PROTOCOL=XOT
         SYSL=DATA=LOGON/00
         TYPE=SPU
         USSTAB=
```

The following display is produced when **DRMT** [**RNM=**]**MCH1** is entered.

```
RMTNAME  OPERAND  TYPE
MCH1     0007E140 MCH
         APPLNAME=A06TSO     (000)
                  CONSOLE    (001)
                  MCHSOL     (002)
         CONNECT=NO
         CTCP=85   (000)
              80   (001)
         CUD=01000000 4C4F4749 4E3A4350 32363231
         CUD0=01   (000)
              FF   (001)
         DCEADDR=2036
         DFLNAME=
         DFXNAME=
         FAC=01014209 09430505
         GATE=GENERAL
         INIT=ACTIVE
         LLC0=NONE
         LLC3=03
              00
         LLC4=04
```

```
LLC5=NONE
LOGTAB=ZZLOGTAB (00006038) LOADED AT 08:53:29 ON 2007/04/09
LUNAME=
MBITCHN=YES
OPTIONS=NOCUD0SELECTSLU
        NODELAYBINDRESP
        ECHODTEADDR
        ECHOFAC
        NOINHIBITBIDREJ
        NOLCN0USED
        NOLLC0CTCPCHK
        NOLLC5CTCPCHK
        MCHTMR=060
        NOONEPIUINB
        NOPFXDCEADDR
        PVCRECONTMR=060
        PVCSETUPREJ=
        PVCSETUPTMR=060
        NOREPDCEADDR
        REQSESSDELAY=002
        NORESETINO
        RESIDSTART=00001
        RTRBIDREJ
        NOSTRIPFAC
        NOSTRIPRTEIN
        SVCCALLTMR=030
PACE=000
PAD=INTEG
PADPARM=001/001
        002/001
        003/002
        004/000
        005/002
        007/021
        012/001
        013/007
        021/000
        110/000
PROTOCOL=XOT
PVC=NONE
PWPROT=YESWOCC
SUBADDR=YES
SUBD=
SVC0=0005
     MCH10001/I/
     MCH10002/I/
     MCH10003/I/
     MCH10004/666666666666666I/
     TEST16LU/1036000301-1036000103-666612345333301O/
SVC3=0015
```

```
     MCH13001/I/
     MCH13101/I/
     Q1990100/I/
     Q2000100/I/
     Q2010100/I/
     Q2020100/I/
     Q2030100/I/
     Q2040100/I/
     Q2050100/I/
     Q2060100/I/
     Q2070100/I/
     Q2080100/I/
     Q2090100/I/
     Q2100100/I/
     Q9910100/I/
SVC4=0002
     MCH14001
     MCH14002
SVC5=0020
     MCH15001/I/
     MCH15002/I/
     MCH15003/I/
     MCH15004/I/
     MCH15005/I/
     MCH15006/I/
     MCH15007/I/
     MCH15008/I/
     MCH15009/I/
     MCH15010/I/
     MCH15011/I/
     MCH15012/I/
     MCH15013/I/
     MCH15014/10360001I/
     MCH15015/I/
     MCH15016/2621I/
     MCH15017/2621I/
     MCH15018/2621I/
     MCH15019/9999I/
     MCH15020/10360001030/MXT1
SYSL=SUBD=01/00
     SUBD=02/00
     SUBD=03/00
     SUBD=04/00
     SUBD=05/02
     SUBD=09/01
     NULL/00
TRAN=SPACE
TYPE=MCH
USSTAB=ISTINCDT (85E00418) LOADED AT 08:54:07 ON 2007/04/09
VCLMT=00048
```

The following display is produced when **DRMT** [**RNM=]MCH1 GATE= USSTAB=** is entered.

```
RMTNAME  OPERAND  TYPE
MCH1     0007E140 MCH
         GATE=GENERAL
         USSTAB=ISTINCDT (85E00418)
```

The following display is produced when **DRMT** [**RNM=]MCH1 SVC0=MCH10020** is entered.

```
RMTNAME  OPERAND  TYPE
MCH1     0007E140 MCH
         SVC0=0020
              MCH15020/1036000103O/MXT1
```

DSTATS Command - Display Statistics

(changed for V2R3M0)

for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**DSTATS**	**[*rmtname*]**	**C**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**DSTATS**			

This command will display HNAS *local* PCE and *global* statistics.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value restricting DSTATS processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, DSTATS processing is restricted to the selected PCEs followed by the global (aggregate) statistics display. If **ID=0** is specified, DSTATS processing is for all PCEs followed by the global (aggregate) statistics display. The global statistics are always displayed last with an omitted PCE ID= value.

  The following is an example of the heading and information produced by the **DSTATS** command.

```
PID  INTERVAL  DISPATCH  TIMEOUT  XMT-PKT  XMT-CHAR  RCV-PKT  RCV-CHAR
|    |         |         |        |        |         |        |
|    |         |         |        |        |         |        |<- number of
|    |         |         |        |        |         |           bytes rcv'd
|    |         |         |        |        |         |
|    |         |         |        |        |         |<- number of
|    |         |         |        |        |            packets rcv'd
|    |         |         |        |        |
|    |         |         |        |        |<- number of bytes xmt'd
|    |         |         |        |
|    |         |         |        |<- number of packets xmt'd
|    |         |         |
|    |         |         |<- number of subtask timeouts
|    |         |
|    |         |<- number of subtask dispatches
|    |
|    |<- delta time (hh:mm:ss)
|
|<- PCE ID number, blank for global
```

The following display is produced when **DSTATS ID=6-9** is entered at the end of the following command sequence:

```
ID=                          <- reset default PCE ID limits
STATS ON                     <- start global stats
STATS ID=6-9 ON              <- start PCE stats for ID=6-9
STATS ID=6-9 OFF            <- stop PCE stats for ID=6-9 after 4 minutes
STATS OFF                    <- stop global stats
DSTATS ID=6-9                <- display results
```

PID	INTERVAL	DISPATCH	TIMEOUT	XMT-PKT	XMT-CHAR	RCV-PKT	RCV-CHAR
0006	0:03:46	00000000	00000000	00000000	00000000	00000000	00000000
0007	0:03:46	00000368	00000001	0000080C	00028343	00000811	000038A5
0008	0:03:46	00000000	00000000	00000000	00000000	00000000	00000000
0009	0:03:46	00000000	00000000	00000000	00000000	00000000	00000000
	0:02:24	00000531	00000004	0000080C	00028343	00000811	000038A5

The following display is produced when **DSTATS ID=7** is entered.

PID	INTERVAL	DISPATCH	TIMEOUT	XMT-PKT	XMT-CHAR	RCV-PKT	RCV-CHAR
0007	0:03:46	00000368	00000001	0000080C	00028343	00000811	000038A5
	0:02:24	00000531	00000004	0000080C	00028343	00000811	000038A5

General notes for DSTATS:

1) The DSTATS command displays the latest statistical information.

2) If statistics have never been started, the interval will display as 0:00:00.

3) If statistics were started (STATS ON) and then stopped (STATS OFF), the interval and counts displayed will be those at the time statistics collecting was terminated.

4) If statistics were started (STATS ON) and left running, the interval and counts displayed will be those at the time of the display.

5) *Global* statistics collecting can also be started when HNAS is activated via the PARM='STATS' option on the Z/OS, OS/390 or MVS program EXEC statement. In the examples above, STATS was specified as an HNAS start option. This is why global statistics are displayed and PCE statistics are not. PCE statistics collecting can only be started using the STATS ON console command.

DTRC Command - Display Trace Table

(changed for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	DTRC	**[*count*]**		

This command will display HNAS trace table entries starting with the most recent entry then going backwards in time.

You may enter a ***count*** value to restrict the display to a specific number of 16-byte trace entries. If no ***count*** value is specified, all trace entries will be displayed.

The following display is produced when **DTRC 16** is entered.

```
CURRENT=002B0710 START=002A4E80 END=002B4E80 WRAPCNT=00000001

002B0710 23242526 2A2B2D3D 3A3B2C2E 2F3F0D34 ................
002B0700 6C6D6E6F 70717273 74757677 78797A40 %_>?
002B06F0 36373839 30616263 64656667 68696A6B .....¬ƒ¿¡ÐÝ«—¶,
002B06E0 7129B9F0 00140047 58595A31 32333435 ….æ0...ÂÏ?!.....
002B06D0 50515253 54555657 00000000 00000000 &ÈÍÎËÌÓÔ........
002B06C0 363A3231 2E313732 20303136 3335204F ..............|
002B06B0 40232425 262A2B2D 3D3A3B2C 2E2F0D34  ..............
002B06A0 6B6C6D6E 6F707172 73747576 7778797A ,%_>?
002B0690 53980008 0C048889 00236A58 8001DD86 ÎQ....HI..¶Ïÿ.
002B0680 00000000 00000000 00000000 00000000 ................
002B0670 00000997 00000000 40404040 40404040 ...P....
002B0660 80019A4C 001300FA 00236BF8 00010000 ÿ.™<...Š..,8....
002B0650 33980008 15462360 00236A58 00019A5A .Q...„.-..¶Ï..™!
002B0640 FFFFFFFF 00000023 00000000 00000000 ****...........
002B0630 0029D9D0 0029D9D0 00000007 00000000 ..R}..R}........
002B0620 00000000 000000C0 0029E53C 00000000 .......{..V.....
002B0610 00000558 00237054 00000010 0029D9FC ...Ï..
002B0600 8001E0D8 00237028 00000024 00237080 ÿ.\Q..
002B05F0 63980008 0C048884 00236A58 8001D768 ƒQ....HD..¶Ïÿ.P«
002B05E0 00000000 00000000 00000000 00000000 ................
002B05D0 00000996 00000000 4CC5E9C1 E2D4C96E ...O....<EZASMI>
002B05C0 80019A4C 00100002 00236C20 00010000 ÿ.™<......%.....
```

DVC Command - Display Virtual Circuit

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

for collection of VCs on a specific MCH

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*mchname*]	DVC	[*mchname*]	C	230
		[{SHOWNAME[{=SLU| =SPU| =MCH| =RTR}]| SHOWADDR}]	N C	230 230
[VCN=*minvcn*[-*maxvcn*]]		[{SHOWCGAD|SHOWCDAD| SHOWIPAD}]	N	230
		[{SHOWVCST|SHOWLUST}]	N	230
		[{SHOWSESSINIT|SHOWCID}]	N	240
		[{FMT1|FMT2}]		
		[STATE={P1|IDLE}| {P2|DTEWAIT}| {P3|DCEWAIT}| {P4D1|DATRDY}| {P4D2|DTERST}| {P4D3|DCERST}| {P5|CALLCOL}| {P6|DTECLR}| {P7|DCECLR}]	N	220
		[TRACE={OFF|ON}]	N	230
		[TYPE={GATE|LLC4}| {PAD|LLC5}| {PCNE|LLC0}| {QLLC|LLC3}]	N	220

or for a collection of VCs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	DVC	[{SHOWNAME[{=SLU| =SPU| =MCH| =RTR}]| SHOWADDR}]	N C	230 230

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[IFN=*minifn*[-*maxifn*]]		**[{SHOWCGAD|SHOWCDAD| SHOWIPAD}]**	**N**	**230**
[VCN=*minvcn*[-*maxvcn*]]		**[{SHOWVCST|SHOWLUST}]**	**N**	**230**
		[{SHOWSESSINIT|SHOWCID}]	N	240
		[{FMT1|FMT2}]		
		[ALLID]		
		[ALLIFN]		
		[ALLVCN]		
		[PROT={XOT|XTP}]		
		[STATE={P1|IDLE}| {P2|DTEWAIT}| {P3|DCEWAIT}| {P4D1|DATRDY}| {P4D2|DTERST}| {P4D3|DCERST}| {P5|CALLCOL}| {P6|DTECLR}| {P7|DCECLR}]	**N**	**220**
		[TRACE={OFF|ON}]	**N**	**230**
		[TYPE={GATE|LLC4}| {PAD|LLC5}| {PCNE|LLC0}| {QLLC|LLC3}]	**N**	**220**

This command will display HNAS VC state information for connected VC's. If only the HEADER line is displayed after the command is entered it indicates that no VC connections exist or nothing matched the display criteria as requested by the command parameters entered or those currently in effect.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting DVC processing to the named MCH only for the specified **VCN=** values. DVC processing is for the VCs on the named MCH only.

- When **VCN=*minvcn*[-*maxvcn*]** is specified, DVC processing is restricted to the selected VCs on the selected MCHs. If **VCN=0** is specified, DVC processing is for all VCs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), DVC processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, DVC processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[*-maxid*]** is specified, DVC processing is restricted to the selected PCEs. If **ID=0** is specified, DVC processing is for all PCEs.

- The **SHOWNAME** operand (the default) requests that the name of the SLU or SPU associated with the VC is to be displayed in the SLUNAME/ADDRESS column. If the VC is LLC0|LLC4|LLC5 (PCNE|GATE|PAD) and does not have an active SLU connection or if the VC is LLC3 (QLLC) and does not have an active SPU connection, no name is displayed.

- The **SHOWNAME=SLU** operand requests that the name of the SLU associated with the VC is to be displayed in the SLUNAME/ADDRESS column. If the VC is LLC0|LLC4|LLC5 (PCNE|GATE|PAD) and does not have an active SLU connection or if the VC is LLC3 (QLLC), no name is displayed.

- The **SHOWNAME=SPU** operand requests that the name of the TYPE=SPU REMOTE definition statement associated with the VC is to be displayed in the SPUNAME/ADDRESS column. This REMOTE identifies the SPU that is using the VC connection. If the VC is not LLC3 or if the VC does not have an active SPU connection (XID sequence incomplete), no name is displayed.

- The **SHOWNAME=MCH** operand requests that the name of the TYPE=MCH|XTP REMOTE definition statement associated with the VC is to be displayed in the MCHNAME/ADDRESS column. This REMOTE identifies the logical MCH that is supporting the VC connection. If the VC does not have an active TCPIP connection, no name is displayed.

- The **SHOWNAME=RTR** operand requests that the name of the TYPE=XOT|XTP REMOTE definition statement associated with the VC is to be displayed in the RTRNAME/ADDRESS column. This REMOTE identifies the router that is supporting the VC connection. If the VC does not have an active TCPIP connection, no name is displayed.

- The **SHOWADDR** operand requests that the memory address of the VC control block (VCB) is to be displayed in the NAME/ADDRESS column.

- The **SHOWCGAD** operand (the default) requests that the ***calling*** DTE address for the VC is to be displayed in the CLGADDR/CLDADDR/IPADDR column.

- The **SHOWCDAD** operand requests that the ***called*** DTE address for the VC is to be displayed in the CLGADDR/CLDADDR/IPADDR column.

- The **SHOWIPAD** operand requests that the IP address for the VC is to be displayed in the CLGADDR/CLDADDR/IPADDR column.

- The **SHOWVCST** operand (the default) requests that the VC state is to be displayed in the VCST/LUST column.

- The **SHOWLUST** operand requests that the SLU state is to be displayed in the VCST/LUST column. If the VC does not have an active SLU connection, no SLU state data is displayed.

- The **SHOWSESSINIT** operand (the default) requests that the direction of the session initiation is to be displayed in the SESSINIT/CID column. If the VC session was established via an

inbound or outbound call, INBOUND or OUTBOUND is displayed.

- The **SHOWCID** operand requests that the VTAM connection identifier for the connected SLU is to be displayed in the SESSINIT/CID column. If the VC does not have an SLU connection, blanks are displayed.

- The **FMT1** and **FMT2** operands (for format 1 and 2) control the type of information that is displayed. Currently, **FMT1=SHOWNAME+SHOWCGAD+SHOWVCST+SHOWSESSINIT** (the default) and **FMT2=SHOWADDR+SHOWIPAD+SHOWLUST+SHOWCID**.

  Note: The SHOW*xxxx* and FMT*x* operands are processed in the order specified. The last value you specify is the winner. For example, if you enter SHOWADDR SHOWIPAD FMT1, SHOWNAME will override SHOWADDR and SHOWCGAD will override SHOWIPAD because FMT1 includes both SHOWNAME and SHOWCGAD.

- The **ALLID** parameter is treated as though RNM= omitted and ID=0 were in effect even if they are not. This means that all PCEs are scanned for their MCH components.

- The **ALLIFN** parameter is treated as though IFN=0 was in effect even if it is not. This means that all MCHs are scanned for their SLU components.

- The **ALLVCN** parameter is treated as though VCN=0 was in effect even if it is not. This means that all VCs are displayed for each MCH subject to filtering by the **PROT=**, **STATE=**, **TRACE=** and/or **TYPE=** operands.

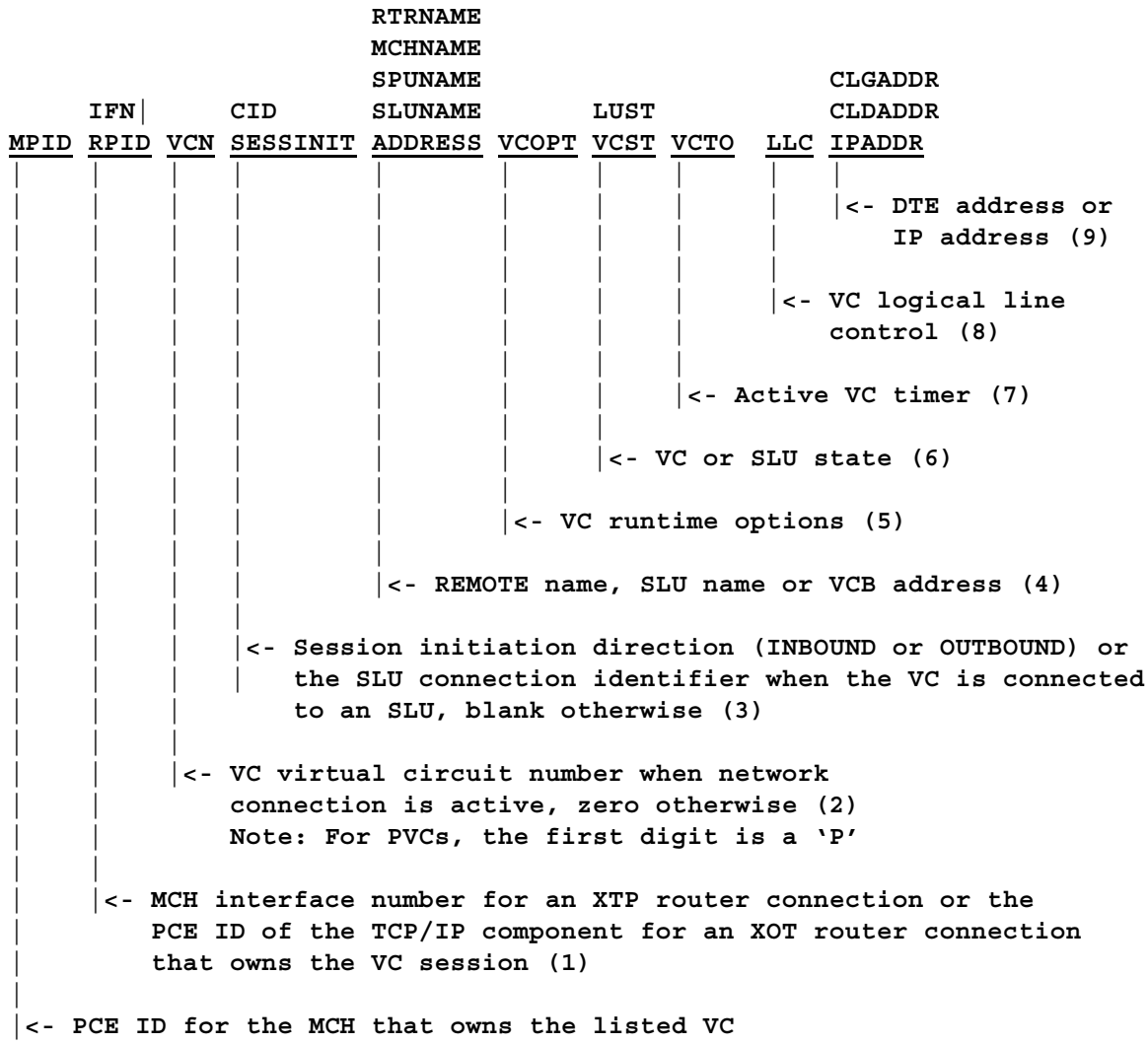
- The **TYPE=** parameter is used to restrict the display to a specific VC type.

- The **PROT=** parameter is used to restrict the display to a specific protocol for **TYPE=TCP** PCEs over which the VCs communicate.

- The **STATE=** parameter is used to restrict the display to a specific VC state.

- The **TRACE=** parameter is used to restrict the display to SLUs being traced (**ON**) or not being traced (**OFF**).

The following is an example of the heading and information produced by the **DVC** command.



Note: MPID and IFN|RPID header display change was introduced into 240 with Enhancement APAR 2400014.

The following display is produced when **DVC ID=8 IFN=3 VCN=1-2** is entered.

```
     IFN|                SPUNAME|
MPID RPID VCN  SESSINIT SLUNAME  VCOPT VCST VCTO  LLC CLGADDR
0008 0003 P001          R1M3P001 V  PN IDLE SUWT  PAD
0008 0003 P002 OUTBOUND R1M3P002 V  PN IDLE       PAD
```

The following display is produced when **DVC BPM SHOWCID SHOWADDR** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
     IFN|
MPID RPID VCN  CID      ADDRESS  VCOPT VCST VCTO  LLC CLGADDR
0008 0003 P001 00000000 0034BB38 V  PN DATA       PAD
0008 0003 P002 00000000 0034C060 V  PN DATA       PAD
0008 0004 0001 2000000C 002B6248 V  SN DATA       PAD 8103600030000000
03A2 000A P001 6A000008 0044D310 V  PN DATA       PAD
03A2 000B P002 4C00000A 0044D838 V  PN DATA       PAD
03A2 0398 0001 30000006 002B6008 V  SN DATA       PAD A103600010400000
03A2 0399 0001          002B6128 V  SN DATA      CONS A103600010300000
```

The following display is produced when **DVC BPM SHOWCID SHOWLUST SHOWCDAD** is entered. Note that the display lists the state information in PCE creation order, not PCE ID order.

```
     IFN|                SPUNAME|
MPID RPID VCN  CID      SLUNAME  VCOPT LUST VCTO  LLC CLDADDR
0008 0003 P001 00000000 R1M3P001 V  PN IDLE       PAD
0008 0003 P002 00000000 R1M3P002 V  PN IDLE       PAD
0008 0004 0001 2000000C R1M45003 V  SN  DTA       PAD 8203600020000000
03A2 000A P001 6A000008 MCH1P001 V  PN  DTA       PAD
03A2 000B P002 4C00000A MCH1P002 V  PN  DTA       PAD
03A2 0398 0001 30000006 MCH15001 V  SN  DTA       PAD 8203600020000000
03A2 0399 0001                   V  SN           CONS 8203600090000000
```

General notes for DVC:

1) The **IFN|RPID** column identifies either the physical MCH link on an IBM XTP router (REMOTE TYPE=XTP,IFNUM=xx,...) or the PCE ID of the TCP/IP component for a Cisco XOT router (REMOTE TYPE=XOT,...) to which the VC is connected.

2) The **VCN** column identifies the HNAS Virtual Circuit Number. For PVCs only, the first digit of the VCN will be replaced by the character 'P'.

For XTP, the VCN is the index into the MCH Address Vector Table (AVT) for the VC. The AVT is used to locate the VC control block when a packet is sent or received across the TCPIP socket connection.

For XTP PVCs, the index is the same as the Circuit Identifier carried in the PVC Info packet. The same Circuit Identifier will be used in all subsequent packets sent and received.

For XTP Callout SVCs, the index is the same as the Circuit Identifier carried in the Outbound Call Request packet created by HNAS. The index is created from the first available slot in the AVT starting at the end and working backwards.

For XTP Callin SVCs, the index is assigned based on an available slot in the AVT starting at the beginning and working forward. The index is not related to the Circuit Identifier received from the router.

For XOT, this is an internal value used for information only.

For XOT PVCs, the VCN is the same as the LCN supplied in the PVC= operand which will be the LCN carried in the PVC Setup packet and all subsequent packets sent and received.

For XOT Callout SVCs, the VCN is always set to 0001 which is then used as the LCN carried in the Outbound Call Request packet created by HNAS and all subsequent packets sent and received.

For XOT Callin SVCs, the VCN is the LCN that is received from the router in the Inbound Call Request packet. It will be used as the LCN in all subsequent packets sent and received.

3) The **SESSINIT/CID** column identifies either how the VC session was established, that is, by an INBOUND or OUTBOUND call (SHOWSESSSINIT) or the VTAM connection identifier when the VC has an SLU component that is bound.

4) The ***xxx*NAME/ADDRESS** column identifies either SLU name (SHOWNAME or SHOWNAME=SLU), a TYPE=SPU REMOTE definition statement name (SHOWNAME or SHOWNAME=SPU), a TYPE=MCH|XTP REMOTE definition statement (SHOWNAME=MCH), a TYPE=XOT|XTP REMOTE definition statement name (SHOWNAME=RTR) or the VCB address (SHOWADDR). An SLUNAME is displayed only if the VC is LLC0|LLC4|LLC5 and has an active SLU connection. An SPUNAME is displayed only if the VC is LLC3 and has an active SPU connection (XID sequence is complete). A RTRNAME|MCHNAME is displayed only if the VC has an active TCPIP connection.

5) The **VCOPT** column identifies HNAS options that are currently active for the VC. A blank indicates that the option is not active. The VCOPT column also displays VC connect status information. The following options can be displayed.

```
VD.SN
|||||<- N => SVC is attached to net or PVC completed setup
||||    D => SVC is detached from net or PVC has not completed setup
||||    I => PVC has received link inoperative reset (1D/73)
||||
||||<-- S => Switched VC (SVC)
|||     P => Permanent VC (PVC)
|||
|||<--- reserved
||<---- D => TRCVC DBK
|<----- V => TRCVC ON
```

6) The **VCST/LUST** column identifies the either current VC state (SHOWVCST) or current SLU state (SHOWLUST).

The following VC states can be displayed for SHOWVCST.

CALX	Call Request transmitted (P2)	**CALR**	Call Request received (P3)
CLRX	Clear transmitted (P6)	**CLRR**	Clear received (P7)
COLL	Call/Clear collision (P5)	**DATA**	VC is active (P4D1)
IDLE	VC is inactive (P1)	**RSTX**	Reset transmitted (P4D2)
RSTR	Reset received (P4D3)		

The following SLU states can be displayed for SHOWLUST.

BN	PLU BIND issued	**BND**	LU is bound
CLR	PLU CLEAR issued	**DTA**	LU data traffic active
IDLE	LU is inactive	**RQS**	SLU REQSESS issued
SD	PLU SHUTD issued	**SDC**	SLU SHUTC issued
SDT	PLU SDT issued	**UNB**	PLU UNBIND issued
UBNS	SLU UNBIND issued		

7) The **VCTO** column identifies the active timer if one is running. The following timer values can be displayed.

CALL	SVC Call Request sent to DTE, response expected	**CLR**	SVC Clear Request sent to DTE, response expected
CLSD	QLLC SPU close wait state, open performed on timer expiration	**DISC**	QLLC QDISC sent to SPU, response expected
EMSG	Datafono EMSGE sent to DTE, response expected	**ICOL**	QLLC SPU input collection in progress
MSYN	Datafono MULTISYNC sent to DTE, response expected	**OMTR**	QLLC SPU output metering in progress
REXP	QLLC PIU sent to SPU, response expected	**RQSD**	REQSHUT sent to VTAM, SHUT-DOWN expected
RQSE	REQSESS sent to VTAM, BIND expected	**RQWT**	PVC reconnect wait state (PVCRE-CONTMR=), REQSESS issued on timer expiration
STLG	REQSESS wait state (REQSESS-DELAY=), REQSESS issued on timer expiration	**STMD**	QLLC QSM sent to SPU, response expected
STUP	PVC Setup sent to DTE, response expected	**SUWT**	PVC Setup wait state (PVC-SETUPTMR=), Setup issued on timer expiration
TEST	QLLC QTEST sent to SPU, response expected	**XID**	QLLC QXID sent to SPU, response expected

Note: The VCTO column was introduced into 240 with Enhancement APAR 2400078.

8) The **LLC** column identifies the VC logical line control. The following LLCs can be displayed.

CONS	Console session	**CON***	Console session (you)
GATE	LLC4 session	**PAD**	LLC5 session
PCNE	LLC0 session	**QLLC**	LLC3 session

Note: If the DVC command is entered from a remote console, an asterisk (*) will replace the 'S' in the 'CONS' text to identify the remote VCB session receiving the display.

9) The **CLGADDR/CLDADDR/IPADDR** column identifies the either ***calling*** DTE address (SHOWCGAD), ***called*** DTE address (SHOWCDAD) or IP address (SHOWIPAD).

The first digit of a DTE address is the address length which is given in hexadecimal notation. The remaining digits are decimal.

For a **PVC**, DTE addresses come from the PVC Information packet (XTP) or PVC Setup packet (XOT). For an SVC, DTE addresses come from the Call Request packet.

ECHOXEQ Command - HNASXEQ Echo Control

(added for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	ECHOXEQ	**[{MAX|MIN|NO}]**	**A**	**240**

This command is used to control how much modifier and parameter data is echoed back to the console operator prior to command execution. Echoed data is prefixed by the HNASXEQ-> text string.

- Enter **ECHOXEQ MAX** to cause all applicable command modifiers and parameters to be echoed on multiple lines (if necessary) with the HNASXEQ-> prefix for each line. Data at the end of one line is continued on the next with an ellipsis (...) suffix as an indication that more information exists.

- Enter **ECHOXEQ MIN** to cause all applicable command modifiers and parameters to be echoed on a single line. If more information exists than can be displayed on one line, an ellipsis (...) will terminate the line. ECHOXEQ MIN is the default state.

- Enter **ECHOXEQ NO** to cause the echoing of command modifiers and parameters to be inhibited.

Note: ECHOXEQ support was introduced into 240 with Enhancement APAR 2400014.

To summarize, you may enter a ECHOXEQ parameter that requests a specific action to be performed as follows:

ECHOXEQ Action Matrix

PARAMETER	ACTION
MAX	**Force all command modifier and parameter information to be echoed on multiple lines, if necessary, prior to command execution.**
MIN or omitted	**Force a single line of command modifier and parameter information to be echoed prior to command execution.**
NO	**Inhibit command modifier and parameter echoing prior to command execution.**

Example 1: Display REMOTE definitions for MCH1 with ECHOXEQ MIN in effect.

```
HNASCMD-> DRMT MCH1
HNASXEQ-> DRMT RNM=MCH1 APPLNAME CONNECT CTCP CUD CUD0 DCEADDR DF...
RMTNAME  OPERAND  TYPE
MCH1     00831EAC MCH
         APPLNAME=MCHSOL     (000)
         :
```

Example 2: Display REMOTE definitions for MCH1 with ECHOXEQ MAX in effect.

```
HNASCMD-> ECHOXEQ MAX
HNASXEQ-> ECHOXEQ MAX
HNASCMD-> DRMT MCH1
HNASXEQ-> DRMT RNM=MCH1 APPLNAME CONNECT CTCP CUD CUD0 DCEADDR DF...
HNASXEQ->      LNAME DFXNAME DTEADDR FAC GATE HOME IDBLK IDNUM ID...
HNASXEQ->      LETO IFNUM INIT IPADDR LLC0 LLC3 LLC4 LLC5 LOGTAB ...
HNASXEQ->      LUNAME MAXDATA MBITCHN OPTIONS PACE PAD PADPARM PK...
HNASXEQ->      TSIZ PORT PROTOCOL PVC PWPROT SUBADDR SUBD SVC0 SV...
HNASXEQ->      C3 SVC4 SVC5 SYSL TAP TRAN TYPE USSTAB VCLMT
RMTNAME  OPERAND  TYPE
MCH1     00831EAC MCH
         APPLNAME=MCHSOL     (000)
         :
```

General notes for ECHOXEQ:

1) HNAS console sessions activate with **ECHOXEQ MIN** in effect.

END Command - Delimit EXEC Command List

(apar/added for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	END		**A**	**240**

This command is used to delimit an EXEC command list file. When the END statement is encountered, HNAS stops queuing commands. This will reduce the likelihood of receiving the NASC054E message.

EXEC Command - Execute Command List

(new for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	EXEC	**[{LCLCONS|*rconname*}]**	**A**	**240**
		[{*ddname*| **(*options*,** ** *cmd*1,...,*cmd*n)}]** **[LIST *ddname*]** **[STOP]**	**N** **C**	**240** **240**

This command is used to execute a command list file identified by ***ddname*** or supplied as an inline command list **(*cmd*1,...,*cmd*n)**, to display a command list file (**LIST *ddname***) or to stop the execution of a command list that is currently active (**STOP**).

- When the **LCLCONS** or ***rconname*** argument is provided, it directs EXEC processing to the identified console. This allows one console to alter EXEC processing for another console. For example, EXEC RCON0031 (DNAS,ALARM SHOW) will cause the DNAS and ALARM SHOW commands to be executed for the remote console named RCON0031 regardless of which console actually entered the command. For more information on LCLCONS|*rconname* processing, please refer to section entitled 'Console PCE name as a Suffix Command Modifier' on page CONS-11.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

EXEC Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
STOP	**none**	**LCLCONS|*rconname***	**1**
ddname	**(*options*,*cmd*1,...,*cmd*n)**	**LCLCONS|*rconname***	**2**
(*options*,*cmd*1,...,*cmd*n)	***ddname***	**LCLCONS|*rconname***	**3**
LIST	**none**	**LCLCONS|*rconname***	**4**

- Enter **EXEC *ddname*** to execute a command list file identified by *ddname*. *ddname* identifies a DDNAME JCL statement that exists in the HNAS start job. DDNAME, in turn, points a file that contains a list of console commands. The DDNAMEs you that provide in the HNAS start job are arbitrary: For example:

```
//CMDLIST1 DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST1),DISP=SHR
//CMDLIST2 DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST2),DISP=SHR
//CMDLIST3 DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST3),DISP=SHR
```

```
//CMDLIST4 DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST4),DISP=SHR
:
```

These command list files can be executed using the EXEC console command as follows:

```
EXEC CMDLIST1
EXEC CMDLIST2
EXEC CMDLIST3
EXEC CMDLIST4
```

If a *ddname* is specified that does not exist in the HNAS start JCL, the following message is issued:

```
NASC051E EXEC ddname INVALID, CANNOT BE OPENED
```

If a *ddname* identifies a DDNAME that references a member of a partitioned dataset and the member does not exist, the following message is issued:

```
NASC204E ddname ABEND PARMLIST=01301886 0008A8B4 008CC410 00000000
```

When a valid command list *ddname* is specified and it has been read successfully, the following message is generated:

```
NASC055I EXEC ddname FILE HAS BEEN READ, IT WILL NOW BE EXECUTED
```

A command list file can contain another EXEC *ddname* statement. When the embedded EXEC command is decoded, the balance of the current command list will be ignored and the new command list will then be executed. This allows you to chain command lists. If a command list contains an EXEC command that points at itself (e.g. EXEC CMDLIST3 is contained in CMDLIST3), the current command list (CMDLIST3 in this case) will be terminated and the following message will be issued:

```
NASC050E EXEC CMDLIST3 INVALID, CANNOT POINT AT ITSELF
```

You can also chain command lists via JCL by concatenating the command list dataset names as long as the concatenation forms a sequential dataset. For example:

```
//CMDLISTS DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST1),DISP=SHR
//         DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST2),DISP=SHR
//         DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST3),DISP=SHR
//         DD DSN=COMM1.V2R4M0.TEMPMAC(CMDLIST4),DISP=SHR
```

These command list files can be executed using the EXEC console command as follows:

```
EXEC CMDLISTS
```

Note: In the initial development, the command lists are always sequential datasets or members of partitioned datasets. The EXEC command does not currently allow you to specify a DDNAME and MEMBER name if DDNAME references a PDS.

Note: All DDNAMEs and the members they reference are completely arbitrary. The only requirement is that the *ddname* you provide for the EXEC command ***must*** exist in the HNAS start JCL.

Note: If DISP=SHR is specified for all command list DDNAMEs above, it will allow you to modify the command lists while HNAS is running.

The commands you provide in a command list file are completely arbitrary. The EXEC command processor does not check the commands for validity. Validity checking is provided by the individual command processors.

A command list file can contain comment records that start with an asterisk (*) or semi-colon (;) in record column 1. For command records in a command list file, a comment can be provided that follows the command text if the first character of the comment is a semi-colon (just like the CDF parsing). Command text is assumed to start with the first non-blank character of a command record and end with the last non-blank character of the record (prior to the comment start ; if it is present).

If an END statement or embedded EXEC command is decoded, queuing of commands is terminated. Any commands the follow an END statement or EXEC command are ignored.

Queued Command Savearea Size

The maximum number of commands that can be queued depends on the available queued storage space. A total of 512 bytes are reserved for queued console commands. Each blank that is part of a command is counted as a text character. The 512 byte maximum also includes a length byte for each queued command. For example, if EXEC *ddname* referenced a file with DPARM, DMAP and TRCMCH ICR specified on 3 records, 22 bytes would be required to queue the commands: 6 for DPARM (1+5), 5 for DMAP (1+4) and 11 for TRCMCH ICR (1+10). If you provide data in a command list that exceeds the 512 byte limit, the following message will be issued:

```
NASC054W EXEC XXXXXXXX FILE EXCEEDS QUEUED COMMAND LIMIT,
```

Note: The BUILD CONCMDQ= operand queue area is also used to save commands provided by the EXEC and QE (QUITEXEC) console commands.

Note: The BUILD CONCMDQ= operand can include the **EXEC *ddname*** command. If **EXEC *ddname*** is specified in the **CONCMDQ=** operand, it should be the last command in the list. This is because HNAS will stop executing the **CONCMDQ=** command list and start executing the **EXEC *ddname*** command list when it detects the EXEC command (the same memory is used to remember the CONCMDQ= and the EXEC command list values). The BUILD CONCMDQ= command list is considered to be the initial command list.

Note: The DCB parameters for datasets identified by the **EXEC *ddname*** command should be RECFM=FB and LRECL=80 with BLKSIZE= as a multiple of LRECL=.

Listed below is a sample command list file:

```
* THIS IS A SAMPLE CONSOLE COMMAND LIST FILE. COMMENTS CAN
* BE SUPPLIED THAT START WITH '*' OR ';' IN RECORD COLUMN 1.
* COMMENTS ON INDIVIDUAL LINES CAN BE SUPPLIED THAT START
* WITH ';'.  CONSOLE COMMANDS WILL BE EXTRACTED FROM EACH
* RECORD FROM THE FIRST NON-BLANK CHARACTER TO THE LAST
* NON-BLANK CHARACTER ON EACH LINE.  THE SEARCH FOR THE
* FIRST NON-BLANK CHARACTER IS EXECUTED FOR NON-COMMENT
* RECORDS ONLY (*|; IS NOT IN CC1).  THE SEARCH FOR THE
* LAST NON-BLANK CHARACTER IS AFFECTED AFTER THE LINE
* COMMENT STARTING DELIMITER (;) IS FOUND.
* |<-- FIRST NON-BLANK
* |
* |             |<-------- LAST NON-BLANK
* |             |
* V             V
  NOPURGEONERROR              ; NO PURGE ON ERROR OPTION
  DNAS                        ; FIRST QUEUED COMMAND
  DPARM EXEC                  ; SECOND QUEUED COMMAND
```

This file would be processed as though CONCMDQ=(DNAS,DPARM) were specified.

- Enter **EXEC** (***options*,*cmd*1**,...,***cmd*n**) to schedule an inline command list on a single line without having to invoke a ***ddname***. Prior to the introduction of the EXEC command, only multiple non-display type commands could be entered on a single line. Using the EXEC command, multiple display commands can be scheduled. For example, if EXEC (DNAS,DPARM) were entered, the DNAS and DPARM commands would be executed consecutively.

  Currently, ***options*** can be **PURGEONERROR**|**NOPURGEONERROR**

  PURGEONERROR (the default) specifies that the remaining commands in an active command list will be purged if an error occurs for any command in the command list.

  NOPURGEONERROR specifies that the remaining commands in an active command list will still be executed even if an error occurs for any command in the command list.

  Notes: 1) The PURGEONERROR|NOPURGEONERROR options are valid for the EXEC= operand on the BUILD definition statement as well as the EXEC console command.

  2) The PURGEONERROR|NOPURGEONERROR options are not included in the command list but are decoded and acted upon immediately when parsed.

  3) Although the PURGEONERROR|NOPURGEONERROR options should be specified as the first command list element, the parser will accept the keywords anywhere in the list. However, the last occurrence of either option is the value that will be used.

  4) The PURGEONERROR|NOPURGEONERROR option may be specified as the

first non-comment record in a *ddname* command list.

5) The PURGEONERROR|NOPURGEONERROR option remains in effect until changed by the EXEC console command. To view the current option in effect, specify the following:

```
EXEC LIST
```

(command list omitted) which will yield the following display:

```
OPTSONLY COMMAND QUEUE COUNT: 0000/0512
PURGEONERROR <- IN EFFECT
```

6) If the PURGEONERROR|NOPURGEONERROR option is specified and the EXEC *cmdlist* LIST command is entered, the value of the specified option will be listed first regardless of where it was specified in the command list. For example:

```
EXEC (DLP,NOPURGEONERROR,DPARM,DLU) LIST
```

will yield the following display:

```
INLINE  COMMAND QUEUE COUNT: 0014/0512
NOPURGEONERROR <- SPECIFIED
=> DLP
=> DPARM
=> DLU
```

Note: The new NOPURGEONERROR|PURGEONERROR keyword processing was introduced into 240 by Enhancement APAR 2400098.

Note: An inline command list is subject to the same queue criteria as a *ddname* command list. See the section above entitled '**Queued Command Savearea Size**' for details.

- Enter **EXEC *ddname* LIST** to display the individual commands in the command list file identified by *ddname*. If CMDLIST1 identifies a file that contains the sample command list described above, the following would be displayed when EXEC LIST CMDLIST1 was entered:

```
NASC055I EXEC CMDLIST1 FILE HAS BEEN READ, IT WILL NOW BE LISTED

=> DNAS
=> DPARM EXEC
```

Note: EXEC LIST *ddname* displays commands only. Comments are not displayed.

- Enter **EXEC STOP** to terminate the active command list.

Note: Any operator input will terminate a running command and the following message will be issued:

```
NASC005W cmdname ABORTED DUE TO OPERATOR INPUT
```

If the current command is part of a running command list, the following message will also be displayed:

```
NASC006W ddname COMMAND LIST INTERRUPTED
```

The interrupted command list will the resume with the next command. If you wish to stop the active command list, enter EXEC STOP.

General notes for EXEC:

1) The *ddname* you enter must be defined in the EXEC PGM=HNAS start JOB.

2) If the dataset identified by *ddname* contains another EXEC command, the current command list will be terminated and the new one will be started.

3) If an EXEC command that is embedded in a command list points at itself, the current command list will be terminated and the following message will be issued:

```
NASC050E EXEC ddname INVALID, CANNOT POINT AT ITSELF
```

4) Operator commands can be entered while a command list is running but will terminate the current command in progress and then be executed. The remaining commands in the command list will be executed after the operator command completes.

5) You can create a command list that will be executed after it is created by specifying a list of commands, each separated by a comma, within a sublist (parenthesized expression). This will allow the execution of consecutive display commands which cannot be done when commands are executed immediately from normal operator input.

6) Supply an END statement or an embedded EXEC command in a command list file to stop queuing of commands. Any commands that follow the embedded END or EXEC command will be ignored. This will reduce the likelihood of receiving the NASC054E error message due to the 512 byte queuing limit being reached.

7) You may enclose a *cmd*i within single or double quotes (e.g., *cmd*i, '*cmd*i' or "*cmd*i" are valid). If single quotes are part of a *cmd*i, it can be enclosed in double quotes (e.g., smsg 'test' or "smsg 'test'" are valid). The enveloping single or double quotes are removed from each *cmd*i before the values are saved.

8) Leading and trailing blanks are removed from each *cmd*i before the values are saved.

9) For both the EXEC and SCHEDULE command, the CONCMDQ queue is used to execute commands. For the EXEC command, *cmd*i values are enqueued to CONCMDQ in a first in, first out (FIFO) manner. For the SCHEDULE command, *cmd*i values are enqueued to CONCMDQ in a last in, first out (LIFO) manner. LIFO enqueue is used because sched-

uled commands must be executed at the specified time whereas EXEC commands are executed in sequence.

F or FIND Command - Find Data in Storage (Privileged)

(new for V2R1M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	F[IND]	**[B=*startaddress*]**	**N**	**210**
		[E=*endaddress*]		
		D='*textdata*'|*hexdata*		

This command will locate the data given by the **D=** operand within the bounds established by the **B=** and **E=** operands. If the **B=** parameter is omitted, the default base established by the Display Memory or Modify Memory (D base + offset or M base + offset, respectively) will be used. If the **E=** parameter is omitted, the end of HNAS executable memory (NASEND) will be used. The **D=** operand, which is required, may specify up to 63 text characters enclosed in quotes or up to 63 hex bytes without the framing characters X" (hex digits must be paired).

The following display is produced when **F B=6D08 E=FFFF D='XFNASWA'** is entered.

```
CURRENT=00006D08 START=00006D08 END=0000FFFF
FIND=E7C6D5C1 E2E6C1 'XFNASWA'
CURRENT=00007D08 START=00006D08 END=0000FFFF
FIND=E7C6D5C1 E2E6C1 'XFNASWA'
:
```

Note: The FIND command progress is indicated by the value displayed for **CURRENT** parameter. This message is issued each time X'1000' bytes of data has been scanned until the target data is found or **E=** limit is reached.

When the specified data is found, the following message is generated.

```
TEXT FOUND, ADDRESS=00008490
```

This also sets a new base value for the next FIND, DMEM or MMEM command. If you reenter the FIND command with no operands after a 'hit', the command will continue looking for the same data from the new base until the end address specified by the original **E=** operand.

If the specified data is not found within the established boundaries, the following message is produced.

```
TEXT NOT FOUND
```

HELP or ? Command - Display Command Help

(changed for V2R1M0)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	HELP or ?	**{[*command*]|SHOWALL}**	**C**	**240**

This command will display the HNAS Console Subsystem command list with a brief description of each command or the syntax required for a selected command.

You may enter a ***command*** name to restrict the display to a single command. In this case, the display is **verbose** which means that all command information is displayed. If the ***command*** name is omitted, a short list (**non-verbose**) of all commands that are supported by the HNAS Console Subsystem will be displayed. When all commands are displayed, they are listed alphabetically, top to bottom.

When the **SHOWALL** argument is specified, the **verbose** display is produced for every command.

Note: The HELP display will now list the common parameters that may be specified for all commands immediately after the display header record (see example below). This new HELP display support was introduced into 240 with Enhancement APAR 2400014.

The following is an example of the heading and information produced by the **HELP** command.

```
COMMAND  *DESCRIPTION (* => PRIVILEGED)
|        ||
|        ||<- command description
|        |
|        |<- privileged command indicator (*)
|
|<- command name
```

The following display is produced when **HELP TRCADDR** is entered.

```
COMMAND   DESCRIPTION (* => PRIVILEGED)
          COMMON PARAMETERS FOR ALL COMMANDS: cmdname parm
                 parm=? to display help for cmdname.
                      BPM to bypass permanent modifiers for cmdname.

TRCADDR  *EVENT TRACE CONTROL FOR RESOURCE ADDRESSES
          ENTER> [CLDADDR=ddd...ddd] TRCADDR {ADD|DEL|PRG}
          ENTER> [CLGADDR=ddd...ddd] TRCADDR {ADD|DEL|PRG}
          ENTER> [IPADDR=aaa.bbb.ccc.ddd[(ppppp)]] TRCADDR {ADD|DEL|PRG}

          NOTE: TRACING REQUIRES ADDITIONAL CPU CYCLES
```

The following display is produced when **HELP** is entered without a command name.

```
COMMAND   DESCRIPTION (* => PRIVILEGED)
          COMMON PARAMETERS FOR ALL COMMANDS: cmdname parm
                 parm=? to display help for cmdname.
                      BPM to bypass permanent modifiers for cmdname.

@         *CUSTOM/SPECIAL COMMANDS FOR COMM-PRO USE ONLY
ALARM     *ALARM CONSOLE CONTROL
CID=       SET LOGICAL UNIT CONNECTION IDENTIFIER EXTENTS
CLDADDR=  SET RESOURCE CALLED DTE ADDRESS
CLGADDR=  SET RESOURCE CALLING DTE ADDRESS
DADDR      DISPLAY RESOURCE ADDRESSES
D[MEM]     DISPLAY MEMORY
DLCL       DISPLAY LOCAL CONFIGURATION PARAMETERS
DLP        DISPLAY HNAS LOAD POINT
DLU        DISPLAY LOGICAL UNIT STATUS
DMAP       DISPLAY HNAS MODULE MAP
DMCH       DISPLAY MULTI-CHANNEL LINK STATUS
DNAS       DISPLAY HNAS INFORMATION
DPARM      DISPLAY CONSOLE PARAMETER VALUES
DPCE       DISPLAY PROCESS CONTROL ELEMENT STATUS
DRMT       DISPLAY REMOTE CONFIGURATION PARAMETERS
DSTATS     DISPLAY PROCESS CONTROL ELEMENT STATISTICS
DTRC       DISPLAY HNAS TRACE TABLE
DVC        DISPLAY VIRTUAL CIRCUIT STATUS
ECHOXEQ    MODIFY COMMAND EXECUTION ECHO OPTION
EXEC       EXECUTE COMMAND LIST
F[IND]    *FIND DATA IN MEMORY
HELP       DISPLAY CONSOLE COMMAND HELP INFORMATION
ID=        SET PROCESS CONTROL ELEMENT IDENTIFIER
IFN=       SET MULTI-CHANNEL LINK INTERFACE NUMBER EXTENTS
IPADDR=    SET RESOURCE IP ADDRESS AND OPTIONAL PORT NUMBER
LNM=       SET LOCAL NAME
LNCT=      SET SCROLL LINE COUNT (SIMULATED ATTENTION ENFORCED)
LUN=       SET MULTI-CHANNEL LINK CTCP LOGICAL UNIT EXTENTS
LUNM=      SET LU NAME
M[MEM]    *MODIFY MEMORY
MLCL      *MODIFY LOCAL CONFIGURATION PARAMETERS
MON[ITOR] MONITOR RESOURCE UTILIZATION
MRMT      *MODIFY REMOTE CONFIGURATION PARAMETERS
PAUSE      PAUSE CONSOLE PROCESSING (GO TO SLEEP)
PFXWTO    *WTO PREFIX CONTROL
PING      *PING REMOTE ROUTER AND/OR MCH LINK
PRNT      *CONSOLE PRINT CONTROL
Q[UIT]     TERMINATE CONSOLE OPERATIONS
RMTCONS   *REMOTE CONSOLE ACCESS CONTROL
RNM=       SET REMOTE NAME
SCHEDULE   SCHEDULE COMMAND LIST
SHOW      *CONSOLE LOGGING CONTROL
SMSG       SEND MESSAGE TO CONSOLE OPERATOR
SNAP      *SNAP A DUMP OF SELECTED HNAS AREAS
STATS     *STATISTICS CONTROL FOR PROCESS CONTROL ELEMENTS
TRCADDR   *EVENT TRACE CONTROL FOR RESOURCE ADDRESSES
```

```
TRCALL   *TRACE ALL CONTROL
TRCBFR   *BUFFER TRACE CONTROL FOR PROCESS CONTROL ELEMENTS
TRCBFRQ  *BUFFER REQUEST TRACE CONTROL
TRCBST   *BINARY SEARCH TABLE TRACE CONTROL
TRCCNFG  *CONFIGURATION UPDATE TRACE CONTROL
TRCDATA  *DATA TRACE CONTROL FOR PROCESS CONTROL ELEMENTS
TRCDBK   *DATA BLOCK TRACE CONTROL FOR LOGICAL UNITS AND VIRTUAL CIRCUITS
TRCDISP  *DISPATCHER TRACE CONTROL FOR PROCESS CONTROL ELEMENTS
TRCIO    *I/O TRACE CONTROL FOR PROCESS CONTROL ELEMENTS
TRCLU    *EVENT TRACE CONTROL FOR LOGICAL UNITS
TRCLUQ   *EVENT TRACE CONTROL FOR LOGICAL UNIT QUEUES
TRCMCH   *EVENT TRACE CONTROL FOR MULTI-CHANNEL LINKS
TRCMCHX  *EVENT TRACE CONTROL FOR MULTI-CHANNEL LINK GATE LOGICAL UNITS
TRCPCE   *EVENT TRACE CONTROL FOR PROCESS CONTROL ELEMENTS
TRCPRNT  *PRINT TRACE CONTROL
TRCSUBR  *SUBROUTINE CALL TRACE CONTROL
TRCTASK  *TASK WAIT/POST TRACE CONTROL
TRCTRAP  *TRACE TRAP CONTROL
TRCVC    *EVENT TRACE CONTROL FOR VIRTUAL CIRCUITS
TRCVCQ   *EVENT TRACE CONTROL FOR VIRTUAL CIRCUIT QUEUES
TRCWTO   *WTO CALL TRACE CONTROL
V[ARY]   *VARY RESOURCE STATE
VCN=      SET VIRTUAL CIRCUIT EXTENTS
```

ID= Modifier - Set PCE Identifier

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
ID=*minid*[-*maxid*] **ID=?**	***command***	**[ID=*minid*[-*maxid*]]**	**C**	**240**

This modifier is used to select specific PCE(s) for the commands that follow. HNAS allocates a PCE for every process that it controls. The **ID=** value serves as an identifier for each HNAS subtask. For more information on PCE identifiers, refer to page CONS-33 of this document.

Note: The ID= modifier is used by the DLU, DMCH, DPCE, DSTATS, DVC, MON, SMSG, STATS, TRCBFR, TRCCONS, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ and VARY commands.

- If ID= (null) is specified, the command operates on all PCEs unless noted otherwise.
- If ID=0 is specified (ID=0-0 is not permitted), the command also operates on all PCEs unless noted otherwise. For most commands, ID= (null) and ID=0 are treated identically. The exceptions are for the TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ, VARY and MON TAP commands. Please refer to description of these commands for further details.
- If an ID= value is specified as a range (e.g., ID=1-3), the command operates on the selected PCEs.
- If an ID= value is specified as a single element (e.g., ID=1) or as a range where the extents are the same (e.g., ID=1-1), the command operates on the selected single PCE.
- If an invalid ID= value is specified, the balance of the command line is ignored.
- If an ID= value is specified, it must be entered as a hexadecimal number.

  ID=*minid*[-*maxid*] extent values can range from 1 to *n* where *n* is the last configured PCE. To see the value of *n*, issue the DPARM command and look under the VALID VALUES column for the ID= modifier.

- Enter **ID=?** to display the permanent ID= value.

IFN= Modifier - Set MCH Interface Number

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
IFN=*minifn*[-*maxifn*] **IFN=?**	***command***	**[IFN=*minifn*[-*maxifn*]]**	**C**	**240**

This modifier is used to select specific XTP MCH(s) for the commands that follow. HNAS allocates an MCH for each TYPE=XTP|MCH REMOTE definition statement. The **IFN=** value serves as an identifier for XTP Multi-channel Link resources only. XOT logical MCHs are identified completely by the **ID=** value. For more information on XTP MCH identifiers, refer to page CONS-35 of this document.

Note: The IFN= modifier is used by the DLU, DMCH, DVC, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCVC and TRCVCQ commands.

- If IFN= (null) is specified, the command operates on all MCHs unless noted otherwise.
- If IFN=0 is specified (IFN=0-0 is not permitted), the command also operates on all MCHs unless noted otherwise. For all commands, IFN= (null) and IFN=0 are treated identically.
- If an IFN= value is specified as a range (e.g., IFN=2-17), the command operates on the selected MCHs.
- If an IFN= value is specified as a single element (e.g., IFN=2) or as a range where the extents are the same (e.g., IFN=2-2), the command operates on the selected single MCH.
- If an invalid IFN= value is specified, the balance of the command line is ignored.
- If an IFN= value is specified, it must be entered as a decimal number.

  IFN=*minifn*[-*maxifn*] values can range from 1 to 255.

- Enter **IFN=?** to display the permanent IFN= value.

IPADDR= Modifier - Set DTE IP Address

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
IPADDR=*aaa.bbb.ccc.ddd*[(*ppppp*)] **IPADDR=?**	***command***	**[IPADDR=*aaa.bbb.ccc.ddd*[(*ppppp*)]]**	**N** **C**	**114** **240**

This modifier is used to select a specific IP address and optional port number for the TRCADDR command. The TRCADDR command is used to start or stop tracing for a specific TCP socket or collection of TCP sockets when no port number is given.

- The **IPADDR** value must be entered in standard 'dotted decimal' notation. If an optional port number is required, it must be enclosed in parentheses and appended to the IP address digits.
- If a null IPADDR value is entered, the remembered IPADDR parameter is reset.
- Enter **IPADDR=?** to display the permanent IPADDR= value.

LNCT= Modifier - Set Remote Console Scroll Line Count

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LNCT=*value* **LNCT=?**	***command***		**N** **C**	**112** **240**

This modifier is used to set the scroll line count for a remote console session. The scroll line count controls how many display lines are written to the remote console terminal before a simulated attention response is required. The simulated attention forces a pause in the display output so that a screen full of data can be easily read before the next screen is presented.

The simulated attention prompt is displayed as 3 asterisks (*******) in the bottom left corner of the display. The operator can enter an 'A' to abort the display or a null line to resume the display. This mechanism allows the remote console operator to control how data is presented to his terminal.

- You may enter a decimal **LNCT=** value between 0 and 65535. A value of zero (0) forces the simulated attention to be inhibited and allows display scrolling to continue unabated. A default **LNCT=** value of **23** is set when the remote console session is established.

- Enter **LNCT=?** to display the current LNCT= value.

LNM= Modifier - Set LOCAL Name

(new for V2R2M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LNM=*lclname* **LNM=?**	***command***	**[LNM=*lclname*]**	**N** **C**	**220** **240**

This modifier is used to select a specific LOCAL definition statement name for the commands that follow. The **LNM=** value must identify a valid LOCAL definition statement in the HNAS Configuration Data File. Use **LNM=** to target a specific server resource.

Note: The LOCAL resource named by the LNM= modifier is used by the DLCL, DNWDF, MLCL and VARY LCL commands.

- When a ***lclname*** value is specified, **LNM=** takes precedence over the **ID=** value.
- If the **LNM=** modifier is entered with no ***lclname*** value (null), the remembered LOCAL name is reset.
- Enter **LNM=?** to display the permanent LNM= value.

LUN= Modifier - Set MCHX (CTCP) Identifier

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LUN=*minlun*[-*maxlun*] **LUN=?**	***command***	**[LUN=*minlun*[-*maxlun*]]**	**C**	**240**

This modifier is used to select specific MCHX(s) for the commands that follow. HNAS allocates an MCHX for each CTCP control session SLU connection. The **LUN=** value serves as an identifier for HNAS CTCP control session SLU resources. For more information on MCHX identifiers, refer to page CONS-35 of this document.

Note: The LUN= modifier is used by the DMCH, TRCLUQ and TRCMCHX commands.

- If LUN= (null) is specified, the command operates on all MCHXs unless noted otherwise.
- If LUN=0 is specified (LUN=0-0 is not permitted), the command also operates on all MCHXs unless noted otherwise. For all commands, LUN= (null) and LUN=0 are treated identically.
- If a LUN= value is specified as a range (e.g., LUN=1-3), the command operates on the selected MCHXs.
- If a LUN= value is specified as a single element (e.g., LUN=1) or as a range where the extents are the same (e.g., LUN=1-1), the command operates on the selected single MCHX.
- If an invalid LUN= value is specified, the balance of the command line is ignored.
- If a LUN= value is specified, it must be entered as a decimal number.

  LUN=*minlun*[-*maxlun*] values can range from 1 to 28.

- Enter **LUN=?** to display the permanent LUN= value.

LUNM= Modifier - Set LU Name

(new for V2R2M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LUNM=*luname* **LUNM=?**	***command***	**[LUNM=*luname*]**	**N** **C**	**220** **240**

This modifier is used to select a specific SLU name for the commands that follow. The **LUNM=** value must identify a valid SLU that is defined in the LUNAME=, PVC=, SVC0= SVC4= or SVC5= operand for some REMOTE definition statement in the HNAS Configuration Data File. Use **LUNM=** to target a specific SLU resource.

Note: The LU resource named by the LUNM= modifier is used by the DLU, TRCLU, TRCMCHX and VARY LU commands.

- When a ***luname*** value is specified, **LUNM=** takes precedence over the **ID=**, **IFN=** and **CID=** or **LUN=** value combination.
- If the **LUNM=** modifier is entered with no ***luname*** value (null), the remembered LU name is reset.
- Enter **LUNM=?** to display the permanent LUNM= value.

M or MMEM Command - Modify Memory (Privileged)

to change the HNAS Maintenance/Use Anniversary Date (EOMDATE)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	M[MEM]	**EOMKEY=*dddddddddddddddd***	**A**	**240**

or to change the HNAS Trial Period Expiration Date (EOTDATE)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	M[MEM]	**EOTKEY=*dddddddddddddddd***	**A**	**240**

or to modify a location in HNAS storage

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	M[MEM]	***address* [+*offset*] *xx...xx***		

This command is used to modify HNAS storage areas.

- **For a permanent distribution**, you may enter **MMEM EOMKEY=*dd...dd*** to allow the MAINTENANCE/USE ANNIVERSARY DATE (EOMDATE) to be extended using a special 16 decimal digit key provided by the ***dd...dd*** digits. HNAS is normally shipped with an EOMDATE that specifies when the MAINTENANCE/USE license will expire. An unexpired EOMDATE is required to use HNAS. In the past, the only way to extend the EOMDATE was to order and install a new refresh distribution. The new MMEM EOMKEY=***dd...dd*** command now allows the EOMDATE to be extened without the need of a new refresh distribution or without having to stop and re-start HNAS. The EOMKEY=*dd...dd* start parameter is also provided so that the EOMDATE can be extended without having to issue the MMEM EOMKEY=*dd...dd* command each time HNAS is stopped and re-started. Please see the PARM='EOMKEY=*dd...dd*' description in Chapter 2 for additional information. The EOMKEY=*dd...dd* digits are provided by Comm-Pro in a special file that is sent as an email attachment or is downloaded from our FTP server. The EOMKEY=*dd...dd* string can then be cut and pasted to the MMEM console command (or the PARM= operand - see Chapter 2). The EOMKEY file has the following format:

```
EOMKEY=7790732630972316
HNAS EOMKEY CREATED AT 14:55:49 ON 2009/09/14
MAINTENANCE/USE ANNIVERSARY DATE IS 2009/12/31
CUSTID=SFD_99999
CUSTINFO=COMM-PRO ASSOCIATES
EMKYDC=3200912311199999
```

Note: MMEM EOMKEY= command logic was introduced into 240 via APAR 2400095.

Note: The EOMKEY= file format was modified via APAR 2400096.

- **For a trial distribution**, you may enter **MMEM EOTKEY=*dd...dd*** to allow the TRIAL PERIOD EXPIRATION DATE (EOTDATE) to be extended using a special 16 decimal digit key provided by the ***dd...dd*** digits. HNAS is normally shipped with an EOTDATE that specifies when the TRIAL AUTHORIZATION will expire. An unexpired EOTDATE is required to use HNAS. In the past, the only way to extend the EOTDATE was to order and install a new trial refresh distribution. The new MMEM EOTKEY=***dd...dd*** command now allows the EOTDATE to be extened without the need of a new refresh distribution or without having to stop and re-start HNAS. The EOTKEY=*dd...dd* start parameter is also provided so that the EOTDATE can be extended without having to issue the MMEM EOTKEY=*dd...dd* command each time HNAS is stopped and re-started. Please see the PARM='EOTKEY=*dd...dd*' description in Chapter 2 for additional information. The EOTKEY=*dd...dd* digits are provided by Comm-Pro in a special file that is sent as an email attachment or is downloaded from our FTP server. The EOTKEY=*dd...dd* string can then be cut and pasted to the MMEM console command (or the PARM= operand - see Chapter 2). The EOTKEY file has the following format:

  ```
  EOTKEY=4961000737880526
  HNAS EOTKEY CREATED AT 08:17:02 ON 2010/11/29
  TRIAL PERIOD EXPIRATION DATE IS 2011/01/28
  CUSTID=SFD_99999
  CUSTINFO=COMM-PRO ASSOCIATES
  ETKYDC=0201101281199999
  ```

  Note: MMEM EOTKEY= command logic was introduced into 240 via APAR 2400106.

- **For a trial distribution**, you may enter **MMEM EOMKEY=*dd...dd*** to change the trial distribution to a permanent distribution dynamically using the key provided by the ***dd...dd*** digits.

  If EOMKEY=4962030747980516 (for example) is used to convert a trial distribution to a permanent distribution, the DNAS display will reflect this by changing DNAS display records 11, 14 an 18 as follows:

  ```
  AUTH=000  SHIPID=1100000011199999 EMKYID=1100000011199999      11
  MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31*                 14
  EOMKEY=4962030747980516 IS IN EFFECT                            18
  ```

  Note that AUTH=000 on DNAS display record 11 above reflects the new permanent status.

  Note: EOMKEY= trial conversion logic was introduced into 240 via APAR 2400106.

- You may enter an absolute ***address*** or a relative address as an offset from an established base (***+ offset***). A new base is set when the ***address*** and ***+ offset*** parameters are entered together. For example, M 6500 + 20 47F0 will modify the contents of address 6520 to 47F0 and set the address base to 6500. The address base is initially set to the NASMAIN load point.

  The hexadecimal digits (***xx...xx***) you enter will overlay the data at the target address. You must enter an even number digits.

  The M command generates the following request message.

```
HNASXEQ-> MMEM ADDRESS=xxxxxxxx DATA=xxxxxxxx xxxxxxxx ... xxxxxxxx
HNASXEQ-> ENTER: N=ABORT, Y=APPLY
```

You may reply with **N** or **Y**.

N causes the modify memory request to be aborted. **Y** causes the modify memory request to be completed. In both cases, a prompt is then issued for a new command.

Note: If the MMEM command references a memory address that HNAS is not authorized to modify (the address is invalid for the machine configuration or the address is not in the same storage key as HNAS), the following error message will be displayed:

```
NASC611W MODIFY MEMORY ADDRESSING EXCEPTION, CONSOLE COMMAND ABORTED
```

MLCL Command - Modify LOCAL Configuration (Privileged)

(new for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LNM=*lclname*	**MLCL**	**[*lclname*]**	**C**	**230**
		[INIT={ACTIVE|IDLE} **[,DELAYTIME=*minutes*]** **[,RETRYLMT=*count*]]**	**N**	**230**
		[OPTIONS= **[,{BALANCERTEIN|** **NOBALANCERTEIN}]** **[,{BALANCERTEOUT|** **NOBALANCERTEOUT}]]**	**N** **A** **N**	**220** **230** **220**
		[RTEIN=[*ith*][,] **{*|CLEAR[-*nth*]|** **SKIP[-*nth*]|** ***mchname*[-*nth*]}** **[/{*|*dteaddr*** **[{T|S}]}]]**	**A** **N** **N** **A**	**230** **240** **240** **230**
		[RTEOUT=[*ith*][,] **{*|*rmtname*[-*nth*]}** **[/{*|*dteaddr*** **[{T|S}]}** **[/{*|*desctext*}]]**	**A** **N**	**230** **240**

This command is used to modify LOCAL configuration parameters for the LOCAL definition statement identified by the **LNM=** modifier. If no **LNM=** value is set, no modification is performed.

To summarize, you may enter a list of LOCAL definition statement operands that you want to modify as follows:

MLCL Action Matrix

PARAMETER	ACTION	XTP	XOT
INIT=	**Supply value for INIT=**		

PARAMETER	ACTION	XTP	XOT
OPTIONS=	Supply value for OPTIONS= [[NO]BALANCERTEIN [[NO]BALANCERTEOUT	 I I	
RTEIN=	Supply value for RTEIN=	I	
RTEOUT=	Supply value for RTEOUT=	I	

Legend: For LOCAL *type* column: I=> Invalid, blank=> supported (see Console Command Parameter Action Matrix Keys description on page CONS-21 for additional information).

General notes for MLCL:

1) The modified parameters will take effect for new sessions only.

2) A new CDF can be created from the original CDF plus any changes that are made during HNAS execution using the MLCL and MRMT console commands. The new CDF is produced when the GENNWDF start parameter is specified and the NEWDEFN DD statement is included in the HNAS start JOB. New or modified records are identified in the new CDF by the characters **;NWDF** starting in character position 67. The NEWDEFN file is maintained in memory until HNAS is stopped at which time it is then written to disk. You can display the updated NEWDEFN file in memory by issuing the DNWDF command which is described on page CONS-89.

MLCL operand syntax:

```
INIT={ACTIVE|IDLE}                                          (XTP|XOT)
     [,DELAYTIME=minutes]
     [,RETRYLMT=count]
```

Specify the LOCAL initialization options to set or reset. If the LOCAL state changes from ACTIVE to IDLE or vice versa, a VARY LCL ON|OFF command is propagated.

Note: Varying a LOCAL resource off and then on again in rapid succession can cause a TCPIP BIND request to fail and the following alarm message to be displayed:

```
NAS2321W BIND REQUEST FAILED RC=FFFFFFFF 00000030
```

The ERNO=30 at the end of the RC indicates that LOCAL 'IP address is in a timed wait because of a LINGER delay from a previous close or another process is using the address'. When this event occurs, HNAS enforces the DELAYTIME= value and will not retry the BIND until the delay timeout expires. At this time, HNAS will also reset the LINGER timeout that the stack is enforcing thus allowing the BIND to complete. If the default DELAYTIME= value

of 5 minutes is to long to wait for a BIND retry, you should specify DELAYTIME=1 in the CDF or modify it using the MRMT INIT=DELAYTIME=1 console command.

```
OPTIONS=[,{BALANCERTEIN|NOBALANCERTEIN}]                          XOT
        [,{BALANCERTEOUT|NOBALANCERTEOUT}]                        XOT
                                                  (changed for V2R3M0)
```

Specify the LOCAL options to set or reset.

For a **TYPE=XOT LOCAL** definition statement, you may specify OPTIONS= operand values of **[NO]BALANCERTEOUT**.

```
RTEIN=[ith][,]{*|CLEAR[-nth]|                                (XOT only)
               SKIP[-nth]|
               mchname[-nth]}
             [/{*|dteaddr[{T|S}]}]
                                                  (changed for V2R4M0)
```

For the ***nth*** occurrence of the named **TYPE=MCH REMOTE** (***mchname***), **SKIP** or **CLEAR** in the RTEIN= operand list on a **TYPE=XOT LOCAL** definition statement, you may add, replace or delete (*****) the decimal DTE address value **(*dteaddr***) which is used as a callin routing identifier.

You may enter up to 15 decimal digits for ***dteaddr***. If you enter a single asterisk (*****) for the callin routing identifier, the old DTE address value is removed. In this case, any DTE address will be able to access an associated MCH.

You may also specify filtering characters for the *dteaddr* value.

@ as the first *dteaddr* character tells HNAS to accept any DTE address whose trailing digits match the *dteaddr* digits that follow the @. For example, @6789 will allow DTE addresses of 123456789, 16789, 3216789, etc. to be considered a match.

***** within the *dteaddr* (not by itself which is recognized as the delete symbol) will be considered a wildcard character. For example, 12**56 will allow DTE addresses of 120056, 120156, 121056, 120256, etc. to be considered a match.

The @ and * characters can be used together.

Note: In the standard US Code Table an **at sign** @ has a value of X'7C'. The equivalent symbol in French is **à** or in German/Italian is **§**.

Note: In the standard US Code Table an **asterisk *** has a value of X'5C'. The equivalent symbol for the EU is the **not-equal symbol**.

You may also add or change the callin routing delimiter (**T|S**) associated with the RTEIN= operand entry.

If the ***nth*** value is omitted, the first occurrence of the ***mchname***, **SKIP** or **CLEAR** in the RTEIN= operand list is used. The ***nth*** value you specify can identify any existing occurrence of ***mchname***, **SKIP** or **CLEAR** in the RTEIN= operand list. If the ***nth*** value exceeds the occurrences for an ***mchname***, **SKIP** or **CLEAR**, a new entry will be created so long as the total number of entries in the RTEIN= operand list does not exceed 1023.

Use the ***ith*** index value to identify an insert point for a new entry (***mchname***) or a delete point for an existing entry (*******). If specified, the ***ith*** value must be given relative to one, that is, 1 identifies the first entry.

Note: Prior to V2R3M0 enhancement APAR 2300056, if a new entry was to be added to the RTEIN= or RTEOUT= operand, it could only be added to the end of each operand's list. If the associated DTE address is a superset of a previous entry, the new entry would never be used. For example, assume the following RTEIN= operand list:

```
RTEIN=(ABCD/47113456,
       EFGH/23456,
       IJKL/345689)
```

When `MLCL RTEIN=XYZA/2345678` is entered, the RTEIN= operand list will look as follows:

```
RTEIN=(ABCD/47113456,
       EFGH/23456,
       IJKL/345689,
       XYZA/2345678)
```

Because DTE address 23456 occurs earlier in the RTEIN= list (for MCH EFGH), the new entry for MCH XYZA will not be accessed.

To correct this anomaly, the MLCL command has been modified so that a new entry can be inserted at a specific position in the RTEIN= and RTEOUT= operand lists. In this way the new entry can be tested before the old entry that has a subset of the new DTE address. Using the example above, to insert XYZA/2345678 as the second entry in the RTEIN= operand list, prefix the MCH name (XYZA) with a decimal entry number value as follows:

```
MLCL RTEIN=2,XYZA/2345678      <- comma before XYZA is optional
```

This command changes the RTEIN= operand list to look as follows:

```
RTEIN=(ABCD/47113456,
       XYZA/2345678,
       EFGH/23456,
       IJKL/345689)
```

Note also that logic has been added that allows you to delete an entire entry, not just its DTE address. To remove the entry for MCH EFGH in the RTEIN= list above, enter the following command:

```
MLCL RTEIN=3,*                  <- comma before * is optional
```

This command changes the RTEIN= operand list to look as follows:

```
RTEIN=(ABCD/47113456,
       XYZA/2345678,
       IJKL/345689)
```

The insert/delete function described above for RTEIN= is also valid for the RTEOUT= operand.

RTEOUT=[*ith*][,]{*|*_rmtname_[-_nth_]} **(XOT only)**
[/{*|*_dteaddr_[{T|S}]}]
[/{*|*_desctext_}]

(changed for V2R4M0)

For the ***nth*** occurrence of the named **TYPE=XOT REMOTE** (***rmtname***) in the RTEOUT= operand list on a **TYPE=XOT LOCAL** definition statement, you may add, replace or delete (*****) the decimal DTE address value **(*dteaddr***) which is used as a callout routing identifier.

You may enter up to 15 decimal digits for ***dteaddr***. If you enter a single asterisk (*****) for the callout routing identifier, the old DTE address value is removed. In this case, any DTE address will be able to access the associated XOT router.

You may also specify filtering characters for the *dteaddr* value.

@ as the first *dteaddr* character tells HNAS to accept any DTE address whose trailing digits match the *dteaddr* digits that follow the @. For example, @6789 will allow DTE addresses of 123456789, 16789, 3216789, etc. to be considered a match.

***** within the *dteaddr* (not by itself which is recognized as the delete symbol) will be considered a wildcard character. For example, 12**56 will allow DTE addresses of 120056, 120156, 121056, 120256, etc. to be considered a match.

The @ and * characters can be used together.

Note: In the standard US Code Table an **at sign** @ has a value of X'7C'. The equivalent symbol in French is **à** or in German/Italian is **§**.

Note: In the standard US Code Table an **asterisk *** has a value of X'5C'. The equivalent symbol for the EU is the **not-equal symbol**.

You may also add or change the callout routing delimiter (**T|S**) associated with the RTEOUT= operand entry.

You may also provide a description that will be used in the NAS7717W message in lieu of the *dteaddr*. The ***desctext*** suboperand must follow the callout routing delimiter (**T|S**) preceded by a forward slash (/). The *desctext* value is a quoted text string that can be from 1 to 15 characters in length including blanks but excluding the quotes.

For example, RTEOUT=(XOTCLNT1/1234T/'NYC DTE #1'). If you enter a single asterisk (*****) or a null text string ('') for the descriptor text, the old *desctext* value is removed. In this case, the DTE address will be used in the NAS7717W message.

If the ***nth*** value is omitted, the first occurrence of the ***rmtname*** in the RTEOUT= operand list is used. The ***nth*** value you specify can identify any existing occurrence of ***rmtname*** in the RTEOUT= operand list. If the ***nth*** value exceeds the occurrences for a ***rmtname***, a new entry will be created so long as the total number of entries in the RTEOUT= operand list does not exceed 1023.

Use the ***ith*** index value to identify an insert point for a new entry (***rmtname***) or a delete point for an existing entry (*****). If specified, the ***ith*** value must be given relative to one, that is, 1 identifies the first entry.

Note: If invalid data is entered for an operand, its value is not changed and the MLCL command is aborted at that point.

MLCL Examples:

As an illustration of how the MLCL and DLCL commands can be used together, consider the following examples. Note that you should always issue a DLCL command before making any changes so you will have a log of initial values.

Initially, the following display is produced when **LNM=LXOT DLCL RTEIN** is entered (LNM=LXOT sets the LOCAL name for subsequent MLCL and DLCL commands).

```
LNM=LXOT DLCL RTEIN

LCLNAME OPERAND TYPE
LXOT     00077084  XOT
         RTEIN=MCH1    /203612345678
               MCH1    /2036
               MCH2    /9
               MCH3    /8
```

1) Change the DTE address for the first occurrence of the MCH named MCH1 from 203612345678 to 8888 then verify the change.

```
MLCL RTEIN=MCH1/8888
DLCL RTEIN

LCLNAME OPERAND TYPE
LXOT     00077084  XOT
         RTEIN=MCH1    /8888
               MCH1    /2036
               MCH2    /9
               MCH3    /8
```

2) Change the DTE address for the second occurrence of the MCH named MCH1 from 2036 to 20360010 then verify the change.

```
MLCL RTEIN=MCH1-2/20360010
DLCL RTEIN

LCLNAME OPERAND TYPE
LXOT     00077084  XOT
         RTEIN=MCH1    /8888
               MCH1    /20360010
               MCH2    /9
               MCH3    /8
```

3) Add a second entry for the MCH named MCH3 with a DTE address of 20360011 then verify the change.

```
MLCL RTEIN=MCH3-2/20360011
DLCL RTEIN
```

```
LCLNAME  OPERAND  TYPE
LXOT      00077084  XOT
          RTEIN=MCH1    /8888
                MCH1    /20360010
                MCH2    /9
                MCH3    /8
                MCH3    /20360011
```

4) Insert a second entry for the MCH named MCH2 with a DTE address of 9903 before the first occurrence of MCH2 in the list then verify the change.

```
MLCL RTEIN=3,MCH2/9903
DLCL RTEIN

LCLNAME  OPERAND  TYPE
LXOT      00077084  XOT
          RTEIN=MCH1    /8888
                MCH1    /20360010
                MCH2    /9903
                MCH2    /9
                MCH3    /8
                MCH3    /20360011
```

MONITOR Command - Monitor HNAS Utilization (Privileged)

(changed for V2R1M0)
(changed for V2R3M0)
(changed for V2R4M0)

to monitor TAP processing for a specific XTP|XOT REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
RNM=*rmtname*	**MON[ITOR]**	**[*rmtname*]**	**A**	**230**
		TAP [{ON|OFF}] **[PKTDATA|MAXDATA|** **MINDATA|NODATA]**	**A** **A**	**230** **240**

or to monitor TAP processing for a collection of XTP|XOT REMOTEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
ID=*minid*[-*maxid*]	**MON[ITOR]**	**TAP [{ON|OFF}]** **[PKTDATA|MAXDATA|** **MINDATA|NODATA]**	**A** **A**	**230** **240**

or to monitor TAP processing for all XTP|XOT REMOTEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	MON[ITOR]	**TAP [{ALLON|ALLOFF}]** **[PKTDATA|MAXDATA|** **MINDATA|NODATA]**	**A** **A**	**230** **240**

or to monitor HNAS global resource utilization

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	MON[ITOR]	**[{*seconds*|10}]**	**N**	**112**
		[{ALL|ALLINIT} |		
		{BFR|BFRINIT}		
		{CPU|CPUINIT}		
		{LU|LUINIT}	**N**	**210**
		{NOTRACE}	**N**	**210**
		{STATS|STATSINIT}		
		{TASK|TASKINIT}	**N**	**210**
		{VC|VCINIT}]	**N**	**210**

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		Terminate Monitoring: Any SYSCONS operation (HNAS command input) will stop MONitor processing for global resource utilization.		

- When the TAP argument is NOT specified, monitoring is for HNAS resource utilization.

  The MON command (non-TAP) is used to start buffer utilization (**BFR|BFRINIT**), CPU utilization (**CPU|CPUINIT**), LU utilization (**LU|LUINIT**), statistics collection (**STATS|STATSINIT**), task utilization (**TASK|TASKINIT**) and/or VC utilization (**VC|VCINIT**).

  The '**INIT**' suffix for the non-TAP arguments cause the associated counts to be reset before the monitor process is started.

  If no resource list is specified, **ALL** is assumed for HNAS *global* resource utilization. Note that entering **MON [ALL]** is equivalent to entering **MON BFR CPU LU STATS TASK VC**.

  When the TAP argument is ***not*** specified, this command will place the console session in monitor mode. The decimal ***seconds*** value specifies the monitor interval **in seconds**. If a ***seconds*** value is not specified, ten (10) seconds is assumed. When the monitor interval expires, results are displayed. A new interval is then started.

  You may specify a ***seconds*** value between 1 and 3600 (1-hour). We do not recommend using a ***seconds*** value less than 10 seconds.

  Note: The ***seconds*** interval is not used when the TAP argument is entered.

  Following the monitor ***seconds*** interval, you may enter the list of resources that you want to monitor. Once started, the monitored resources cannot be changed. If you want to change the monitored resources, simply re-enter the MONitor command with a new resource list.

  If **NOTRACE** is specified, the monitor display will not include a line showing the current trace table pointer prior to each resource display. This argument allows you to reduce the amount of information produced by the monitor display when time interval completes.

  Note: The monitor command for *global* resource utilization is ended by **ANY** operator input.

- When the TAP argument is specified, monitoring is restricted to REMOTE resources.

  The MON TAP command is used to start (**ON**) or stop (**OFF**) *local* PCE TAP monitor processing for the PCEs identified by the command modifiers (ID= or RNM=) or to start (**ALLON**) or stop (**ALLOFF**) *global* PCE TAP monitor processing or to control the type of data (**PKTDATA|MINDATA|MAXDATA|NODATA**) that is displayed for the REMOTEs being monitored. TAP status log entries are written to SYSPRINT only and are identified by alert message IDs

of the form NAS251xM.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting MON TAP ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[*-maxid*]** is specified, MON TAP ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, MON TAP ON|OFF processing is for all PCEs. *The ID= modifier is only used if the RNM= modifier is not set. If ID= is also null, the command is rejected.*

- Enter **MON TAP [ON]** to start *local* PCE TAP monitoring for the PCEs identified by the command modifiers. *Local* PCE TAP monitoring will be activated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **MON TAP OFF** to stop *local* PCE TAP monitoring for the PCEs identified by the command modifiers. *Local* PCE TAP monitoring will be deactivated for all PCEs if **ID=0** and **RNM=** (null) is specified.

  Note: The MON TAP ON|OFF command requires a valid value for either RNM= or ID=. The command will be rejected if both RNM= and ID= are null.

- Enter **MON TAP ALLON** to start *global* PCE TAP monitoring (equivalent to the MONTAP start parameter).

- Enter **MON TAP ALLOFF** to stop *global* and all *local* PCE TAP monitoring.

- Enter **MON TAP PKTDATA** to enable *global* data display with raw packet data being displayed for the routers currently being monitored for TAP.

```
NAS2513M CLIENT=iii.iii.iii.iii(port) SOCKID=sockid PCEID=pceid NAME=rmtname
NAS2513M XOT TAP SEQUENCE status, TRANSMITTING pkttype
NAS2513M PKT=xxxxxxxx...xxxxxxxx                                  <- PKTDATA
```

- Enter **MON TAP MAXDATA** to enable *global* data display with a maximum amount of configuration data being displayed for the routers currently being monitored for TAP.

```
NAS2513M CLIENT=iii.iii.iii.iii(port) SOCKID=sockid PCEID=pceid NAME=rmtname
NAS2513M XOT TAP SEQUENCE status, TRANSMITTING pkttype
NAS2513M DTEADDR=dd...dd DCEADDR=dd...dd                        <- MINDATA|MAXDATA
NAS2513M FAC=xx...xx                                             <- MAXDATA
NAS2513M CUD=xx...xx                                             <- MAXDATA
```

- Enter **MON TAP MINDATA** to enable *global* data display with a minimum amount of configuration data being displayed for the routers currently being monitored for TAP.

```
NAS2513M CLIENT=iii.iii.iii.iii(port) SOCKID=sockid PCEID=pceid NAME=rmtname
NAS2513M XOT TAP SEQUENCE status, TRANSMITTING pkttype
NAS2513M DTEADDR=dd...dd DCEADDR=dd...dd                        <- MINDATA|MAXDATA
```

- Enter **MON TAP NODATA** to disable *global* packet and configuration data displayed for the routers currently being monitored for TAP.

```
NAS2513M CLIENT=iii.iii.iii.iii(port) SOCKID=sockid PCEID=pceid NAME=rmtname
NAS2513M XOT TAP SEQUENCE status, TRANSMITTING pkttype
```

Note: MON TAP PKTDATA|MAXDATA|MINDATA|NODATA argument support was introduced into 240 as Enhancement APAR 2400044.

To summarize, you may enter a MONITOR parameter that requests a specific action to be performed as follows:

MONITOR Action Matrix

PARAMETER	ACTION
ALL or omitted	**Monitor all HNAS resource utilization.**
ALLINIT	**Monitor all HNAS resource utilization, reset counts first.**
BFR	**Monitor HNAS buffer utilization.**
BFRINIT	**Monitor HNAS buffer utilization, reset counts first.**
CPU	**Monitor HNAS CPU utilization.**
CPUINIT	**Monitor HNAS CPU utilization, reset counts first.**
LU	**Monitor HNAS LU utilization.**
LUINIT	**Monitor HNAS LU utilization, reset counts first.**
NOTRACE	**Do not display current trace entry address with MONITOR results.**
STATS	**Monitor HNAS statistics collection.**
STATSINIT	**Monitor HNAS statistics collection, reset counts first.**
TASK	**Monitor HNAS TASK activity.**
TASKINIT	**Monitor HNAS LU activity, reset counts first.**
VC	**Monitor HNAS VC utilization.**
VCINIT	**Monitor HNAS VC utilization, reset counts first.**
TAP	**Provide TAP monitor service.**
Note: To stop monitoring	**Any SYSCONS operation (HNAS command input) will stop MONitor processing for global resource utilization monitoring.**

The following display is produced when **MON 60 ALL** is entered.

```
9:10:33 CPU:      TRACE=001F2EF0
9:10:33 CPU:   INTERVAL=7.220918 (00000006 E2EB6000)
9:10:33 CPU:      TOTAL=40.487110 (00000026 9C8C69BB)
9:10:33 BFR:      TRACE=001F2EF0 AVAILABLE=0000042B MAXIMUM=0000043D
9:10:33 BFR:               00-20%   21-40%   41-60%   61-80%  81-100%
9:10:33 BFR:   INTERVAL=000006D2 00000000 00000000 00000000 00000000
```

```
9:10:33 BFR:      TOTAL=00001AFE 00000000 00000000 00000000 00000000
9:10:33 STATS:    TRACE=001F2EF0
9:10:34 STATS:          DISPATCH  TIMEOUT  XMT-PKT XMT-CHAR  RCV-PKT RCV-CHAR
9:10:34 STATS: INTERVAL=00000271 00000003 0000034D 0001A823 0000034F 00001729
9:10:34 STATS:    TOTAL=00001557 00000091 00000D64 000694EF 00000D63 00005FC9
9:10:34 VC:       TRACE=001F2EF0
9:10:34 VC:               ACTIVE   ATTACH   DETACH
9:10:34 VC:    INTERVAL=         00000000 00000000
9:10:34 VC:       TOTAL=00000007 00000008 00000001
9:10:34 LU:       TRACE=001F2EF0
9:10:35 LU:               ACTIVE   ATTACH   DETACH
9:10:35 LU:    INTERVAL=         00000000 00000000
9:10:35 LU:       TOTAL=00000001 00000002 00000001
9:10:35 TASK:     TRACE=001F2EF0
9:10:35 TASK:              TCPIP     VTAM      MCH      PCE     CONS    TOTAL
9:10:35 TASK:  INTERVAL=0000058D 00000200 0000034C 00000258 00000000 00000D31
9:10:35 TASK:     TOTAL=0000192E 00000892 00000E87 00001562 00000001 000045A9
9:11:36 CPU:      TRACE=001F2EF0
9:11:36 CPU:   INTERVAL=0.193761 (00000000 2F4E1000)
9:11:36 CPU:      TOTAL=40.680871 (00000026 CBDA7F51)
9:11:36 BFR:      TRACE=001F2EF0 AVAILABLE=0000042B MAXIMUM=0000043D
9:11:36 BFR:              00-20%   21-40%   41-60%   61-80%  81-100%
9:11:36 BFR:   INTERVAL=00000001 00000000 00000000 00000000 00000000
9:11:36 BFR:      TOTAL=00001AFF 00000000 00000000 00000000 00000000
9:11:36 STATS:    TRACE=001F2EF0
9:11:37 STATS:          DISPATCH  TIMEOUT  XMT-PKT XMT-CHAR  RCV-PKT RCV-CHAR
9:11:37 STATS: INTERVAL=00000040 00000003 00000000 00000000 00000000 00000000
9:11:37 STATS:    TOTAL=00001597 00000094 00000D64 000694EF 00000D63 00005FC9
9:11:37 VC:       TRACE=001F2EF0
9:11:37 VC:               ACTIVE   ATTACH   DETACH
9:11:37 VC:    INTERVAL=         00000000 00000000
9:11:37 VC:       TOTAL=00000007 00000008 00000001
9:11:37 LU:       TRACE=001F2EF0
9:11:38 LU:               ACTIVE   ATTACH   DETACH
9:11:38 LU:    INTERVAL=         00000000 00000000
9:11:38 LU:       TOTAL=00000001 00000002 00000001
9:11:38 TASK:     TRACE=001F2EF0
9:11:38 TASK:              TCPIP     VTAM      MCH      PCE     CONS    TOTAL
9:11:38 TASK:  INTERVAL=00000005 00000000 00000002 00000040 00000000 00000047
9:11:38 TASK:     TOTAL=00001933 00000892 00000E89 000015A2 00000001 000045F0
```

The following display is produced when **`MON 60 CPU STATS NOTRACE`** is entered.

```
9:10:33 CPU:   INTERVAL=7.220918 (00000006 E2EB6000)
9:10:33 CPU:      TOTAL=40.487110 (00000026 9C8C69BB)
9:10:34 STATS:          DISPATCH  TIMEOUT  XMT-PKT XMT-CHAR  RCV-PKT RCV-CHAR
9:10:34 STATS: INTERVAL=00000271 00000003 0000034D 0001A823 0000034F 00001729
9:10:34 STATS:    TOTAL=00001557 00000091 00000D64 000694EF 00000D63 00005FC9
9:11:36 CPU:   INTERVAL=0.193761 (00000000 2F4E1000)
9:11:36 CPU:      TOTAL=40.680871 (00000026 CBDA7F51)
9:11:37 STATS:          DISPATCH  TIMEOUT  XMT-PKT XMT-CHAR  RCV-PKT RCV-CHAR
9:11:37 STATS: INTERVAL=00000040 00000003 00000000 00000000 00000000 00000000
9:11:37 STATS:    TOTAL=00001597 00000094 00000D64 000694EF 00000D63 00005FC9
```

General notes for MON (without TAP argument):

1) Unlike the MON TAP command, MON (without TAP) locks the console up while the monitor utilization function is running. The monitor utilization function is stopped and the console is restored when any input is entered. The input can be a new command or a null line.

2) The **TRACE=** value in all utilization displays represents the current HNAS trace entry address at the time of the display. It is provided for information only. You can inhibit its display by specifying the **NOTRACE** option.

3) The **INTERVAL=** value(s) in all utilization displays represents usage counts for the current time interval. For the **CPU** utilization display, it represents the amount of CPU processor time (in seconds) that HNAS was given (i.e., its TCB was the active TCB) during the monitor interval. The value in parenthesis is the CPU processor time in time-of-day format.

4) The **TOTAL=** value(s) in all utilization displays represents usage counts since HNAS was loaded or the counts were reset by an 'INIT' request (e.g., BFRINIT). For the **CPU** utilization display, it represents the amount of CPU processor time (in seconds) that HNAS was given (i.e., its TCB was the active TCB) since it was started or re-initialized by a CPUINIT request. The value in parenthesis is the CPU processor time in time-of-day format.

5) The **AVAILABLE=** value in the **BFR** utilization display represents the current number of free buffers in the HNAS buffer pool at the time of the display.

6) The **MAXIMUM=** value in the **BFR** utilization display represents the maximum number of free buffers in the HNAS buffer pool.

7) The **percent** column values **(e.g., 00-20%)** in the **BFR** utilization display represent the number of buffer allocation requests that were made when the buffer pool was depleted by the given percentage of buffers.

The **INTERVAL=** values are counts for the current monitor interval.

The **TOTAL=** values are aggregate counts since they were last reset via the **MON ALLINIT|BFRINIT** command or at HNAS start time.

8) The **counter** values **(e.g., DISPATCH)** in the **STATS** collection display represent the metric for the named statistic.

The **INTERVAL=** values are counts for the current monitor interval.

The **TOTAL=** values are aggregate counts since they were last reset via the **MON ALLINIT|STATSINIT** command, the STATS ON command or via the STATS start parameter.

9) The **counter** values **(e.g., ATTACH|DETACH**) in the **LU and VC** collection display represent the metric for the named statistic.

The **INTERVAL=** values are counts for the current monitor interval. The **ATTACH** and

DETACH counts represent the number of connects and disconnects, respectively, that occurred during the monitor interval.

The **TOTAL=** values are aggregate counts since they were last reset via the **MON ALLINIT|LUINIT** or **ALLINIT|VCINIT** command. The **ACTIVE** column value represents the number of active connections at the time of this display.

10) The **counter** values **(e.g., TCPIP)** in the **TASK** collection display represent the metric for the named statistic. These counts represent the number of times the ECB for the named HNAS component was posted and processed.

The **INTERVAL=** values are counts for the current monitor interval.

The **TOTAL=** values are aggregate counts since they were last reset via the **MON ALLINIT|TASKINIT** command.

General notes for MON (with TAP argument):

1) Unlike the MON (without TAP) command, MON TAP does not lock the console up while the monitor TAP function is running. The monitor TAP function is stopped when MON TAP OFF|ALLOFF is entered (*local*|*global* monitor, respectively). New console commands can always be entered and will be acted upon when the monitor TAP function is running.

2) The TAP monitor will display all packets sent and received during a single TAP interval.

3) The TAP monitor will work for XTP and XOT REMOTEs. For an XOT REMOTE, OPTIONS=TAPWITHCLR can be enabled or disabled.

4) The *global* TAP monitoring state can be displayed with the DPARM EXEC console command.

5) The *global* TAP monitoring state can be changed using the MON TAP {ALLON|ALLOFF} console command.

6) The MON TAP {ALLON|ALLOFF} console command ignores the setting of the RNM= and ID= modifiers.

7) The *local* TAP monitoring state can be displayed with the DPCE console command. In this case, the 5th character under the NASOPT column will show blank if the TAPping PCE is not being monitored or as 'M' if it is.

8) The *local* TAP monitoring state can be changed using the MON TAP {ON|OFF} console command.

9) The MON TAP {ON|OFF} console command uses the setting of the RNM= and/or ID= modifiers to locate the PCEs to be monitored.

10) Most NAS251xM monitor messages contain an English description of the packet that was

transmitted or received as well as the first 24-bytes of the raw packet (starting with the XOT packet length which is 4-bytes). If a packet is less than 24-bytes in length, the display will be truncated.

11) The NAS251*n*M monitor messages are normally written to SYSPRINT only but can also be routed to SYSCONS using the BUILD ALRMFLTR= operand the ALARM FILTER= command. For example, when FILTER=(NAS251*n*M(FC)) is specified, NAS251*n*M monitor messages will flow to SYSCONS if the SHOWON option is in effect. In addition, when FILTER=(NAS251*n*M(FU)) is specified, NAS251*n*M monitor messages will flow to SYSCONS unconditionally.

12) The NAS251*n*M monitor messages can be filtered from SYSPRINT and/or SYSCONS by the BUILD ALRMFLTR= operand or the ALARM FILTER= command. For example, when FILTER=(NAS251*n*M(P)) is specified, NAS251*n*M monitor messages will be purged from SYSPRINT and SYSCONS.

13) The NAS251*n*M monitor messages are in addition the existing Keep Alive alarm messages (NAS2501W, NAS2502E, NAS2503W) which are written to both SYSPRINT and the SYSCONS.

Note: The MON TAP ON console command causes special monitor messages to be written to SYSPRINT when the PRNT ON option is in effect. These special monitor messages start with a message ID of NAS251*n*M and are not normally sent to the SYSCONS (system operator console). However, these messages can be forced to go to SYSCONS by specifying the message IDs in the ALRMFLTR= list either via the CDF or the ALARM command.

For example, if ALRMFLTR=(...,NAS251*M(FU),...) is specified on the BUILD definition statement or if the ALARM FILTER=(...,NAS251*M(FU),...) console command is issued, TAP monitor messages will also go to SYSCONS.

Please refer to the specific **NAS251*n*M** message type located in the Alert Messages section of the HNAS Messages and Codes Guide.

MRMT Command - Modify REMOTE Configuration (Privileged)

(changed for V2R1M0)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
RNM=*rmtname*	**MRMT**	**[*rmtname*]**	**C**	**230**
		[CUD={NULL|NONE|[*nn/*]*xx...xx*}]	**N** **C**	**114** **210**
		[DCEADDR={NULL|NONE|*dd...dd*}]		
		[DTEADDR={NULL|NONE|*dd...dd*}]	**N**	**230**
		[FAC={NULL|NONE|[*nn/*]*xx...xx*}]	**C**	**210**
		[IDBLK=*xxx*]	**N**	**220**
		[IDNUM=*xxxxx*]	**N**	**220**
		[IDLETO=*minutes*]	**N**	**230**
		[INIT={ACTIVE|IDLE}]	**N**	**230**
		[IPADDR=*a.b.c.d*[(*port*)]]	**N**	**240**
		[LOGTAB=*lgtbname*[-R]]	**A** **C**	**210** **240**
	For MCH	**[LUNAME=*sluname*** **[-{A|I}]** **[/{*pluname*{*}|*}]]**	**A**	**240**
	For DFS	**[LUNAME=*sluname*** **[-{A|I}]** **[/{*pluname*]** **[/{*idnum*|*}]]**	**A**	**240**
	For DFL	**[LUNAME=*sluname*** **[-{A|I}]** **[/{*pluname*|*}]]**	**A**	**240**
	For SPU	**[LUNAME=*sluname*** **[-{A|I}]** **[/{*rpcecnt*|*}]** **[/{*spcecnt*|*}]** **[/{*applid*|*}]** **[/{*mxtname*|*}]]**	**N** **A** **N** **N**	**220** **240** **230** **230**

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	For SPU	**[LUNAME=(*locaddr*,** **{*sluname*|*}** **[-{A|I}]** **[/{*rpcecnt*|*}]** **[/{*spcecnt*]|*}]** **[/{*applid*|*}]** **[/{*mxtname*|*}],** **:** **{*sluname*|*}** **[-{A|I}]** **[/{*rpcecnt*|*}]** **[/{*spcecnt*]|*}]** **[/{*applid*|*}]** **[/{*mxtname*|*}])]**	**N** **A** **N** **N** **A** **N** **N**	**230** **240** **230** **230** **240** **230** **230**
		[MAXDATA=*bytecnt*]	**N**	**220**
		[MBITCHN={YES|NO}]	**N**	**240**

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[OPTIONS=	A	210
		[CLOTINITYP={NONE|BIND|TIMER| CONSOLE}]	A	230
		[CLOTFAILRTYLMT=*count*]	A	230
		[CLOTCONLMT=*count*]	A	230
		[,{CLOSEONTAPFAILURE| NOCLOSEONTAPFAILURE}]	A	240
		[,{CUD0SELECTSLU| NOCUD0SELECTSLU}]	A	240
		[,{DATAF|NODATAF]	N	240
		[,{DATAFAM|NODATAFAM}]	N	240
		[,{DELAYBINDRESP| NODELAYBINRESP}]	N	240
		[,{ECHODTEADDR| NOECHODTEADDR}]	A	220
		[,{ECHOFAC|NOECHOFAC}]	A	220
		[,{EMSGE|NOEMSGE}]	N	240
		[,{IDTST|NOIDTST}]	A	240
		[,{IMS|NOIMS}]	N	240
		[,{INHIBITBIDREJ| NOINHIBITBIDREJ}]	N	230
		[,{LLC0CTCPCHK| NOLLC0CTCPCHK}]	A	230
		[,{LLC5CTCPCHK| NOLLC5CTCPCHK}]	A	230
		[,MCHTMR=*seconds*]	A	211
		[,NRITAB=*nrtbname*[-R]]	N	240
		[,{ONEPIUINB|NOONEPIUINB}]	N	230
		[,{PEER|PRI|SEC}]	N	220
		[,{PFXDCEADDR|NOPFXDCEADDR}]	A	230
		[,PVCRECONTMR=*seconds*]	A	240
		[,PVCSETUPTMR=*seconds*]	A	240
		[,PVCSETUPREJ=*code*]	A	240
		[,{REPDCEADDR|NOREPDCEADDR}]		
		[,REQSESSDELAY=*seconds*]	N	230
		[,{RESETINO|NORESETINO}]	A	240
		[,{RETPIU|NORETPIU}]	N	240
		[,{RTRBIDREJ|NORTRBIDREJ}]	N	230
		[,{STRIPFAC|NOSTRIPFAC}]		
		[,{STRIPRTEIN|NOSTRIPRTEIN}]		
		[,SVCCALLTMR=*seconds*]	A	240
		[,{TAPWITHCLR|NOTAPWITHCLR}]	A	220
		[,TCPRBLMT=*bfrcnt*]	A	220
		[,XID={NO|STD| TAB|TABSTD| (*idnum*,*idcnt*)}]]	N	240
		[PACE={NONE|NULL| *pktcnt*}| ({*rpcecnt*|*} ,{*spcecnt*|*})]	A	220

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
		[PADPARM=NULL|NONE| **(*pn*1,*pv*1,...** **,*pn*k,*pv*k)]**	**A**	**210**
		[PORT=*number*]	**N**	**240**
		[PROTOCOL={XTP|XOT}]	**N**	**240**
		[SVC0=*sluname*=[{*newname*|*}] **[+*gluname*][-{A|I}]** **[/{*dteid*1|*}** **[-{*dteid*2|*}** **[-{*dteid*3|*}]]** **{T|O|I}[{*applid*|*}]]** **[/{*mxtname*|*}]** **[/{*cud*|*}]]** ***dteid*=*dteaddr*|X*idnum*|<*rmtname*>**	**C** **A** **A** **C**	**220** **240** **230** **240**
		[SVC3=*spuname* **[/{*dteaddr*|*}[{I|O}]]** **[/{*mxtname*|*}]]**	**A** **C**	**220** **230**
		[SVC4=*sluname*=[{*newname*|*}] **[-{A|I}]]**	**A** **A**	**230** **240**
		[SVC5=*sluname*=[{*newname*|*}] **[+*gluname*][-{A|I}]** **[/{*dteid*1|*}** **[-{*dteid*2|*}** **[-{*dteid*3|*}]]** **{T|O|I}[{*applid*|*}]]** **[/{*mxtname*|*}]** **[/{*cud*|*}]]** ***dteid*=*dteaddr*|X*idnum*|<*rmtname*>**	**C** **A** **A** **C**	**220** **240** **230** **240**
		[TAP=*value*]	**A**	**114**
		[TRAN={NO|USER| **EVEN|ODD|** **MARK|SPACE|** **NPSIEVEN|NPSIODD|** **NPSIMARK|** **NPSISPACE}]**	**A**	**230**
		[USSTAB=*ustbname*[-R]]	C	240

This command is used to modify REMOTE configuration parameters for the REMOTE definition statement identified by the **RNM=** modifier. If no **RNM=** value is set, no modification is performed.

To summarize, you may enter a list of REMOTE definition statement operands that you want to modify as follows:

MRMT Action Matrix

PARAMETER	ACTION	XTP	XOT	MCH	MXT	SPU	DFL	DFS	DFX	DMY	SVC
CUD=	Supply value for CUD=						I	I	I		
DCEADRR=	Supply value for DCEADDR=						I	I	I		
DTEADDR=	Supply value for DTEADDR=	I		I			I	I	I		
FAC=	Supply value for FAC=						I	I	I		
IDBLK=	Supply value for IDBLK=	I	I	I	I		I	I	I	I	I
IDNUM=	Supply value for IDNUM=	I	I	I	I		I	I	I	I	I
IDLETO=	Supply value for IDLETO=		I				I	I	I	I	I
INIT=	Supply value for INIT=				I	I	I	I	I	I	I
IPADDR=	Supply value for IPADDR=	I	I	I	I	I	I	I	I		I
LOGTAB=	Supply value for LOGTAB=		I				I	I	I	I	I
LUNAME=	Supply value for LUNAME=		I		I				I	I	I
MAXDATA=	Supply value for MAXDATA=	I	I	I	I		I	I	I	I	I
MBITCHN=	Supply value for MBITCHN=		I		I	I	I	I	I	I	I

PARAMETER	ACTION	XTP	XOT	MCH	MXT	SPU	DFL	DFS	DFX	DMY	SVC
OPTIONS=	Supply value for OPTIONS=										
	CLOTINITYP=	I	I	I	I		I	I	I	I	I
	CLOTFAILRTYLMT=	I	I	I	I		I	I	I	I	I
	CLOTCONLMT=	I	I	I	I		I	I	I	I	I
	[NO]CLOSEONTAPFAILURE			I	I	I	I	I	I	I	I
	[NO]CUD0SELECTSLU		I			I	I	I	I	I	I
	[NO]DATAF	I	I	I	I	I	I	I		I	I
	[NO]DATAFAM	I	I		I	I	I	I		I	I
	[NO]DELAYBINDRESP		I		I	I	I	I	I	I	I
	[NO]ECHODTEADDR	I	I		I	I	I	I	I	I	I
	[NO]ECHOFAC	I	I		I	I	I	I	I	I	I
	[NO]EMSGE	I	I	I	I	I	I	I		I	I
	[NO]IDTST	I	I	I	I	I	I	I		I	I
	[NO]IMS	I	I	I	I	I	I	I		I	I
	[NO]INHIBITBIDREJ		I		I	I	I	I	I	I	I
	[NO]LLC0CTCPCHK	I	I		I	I	I	I	I	I	I
	[NO]LLC5CTCPCHK	I	I		I	I	I	I	I	I	I
	MCHTMR=		I		I	I	I	I	I	I	I
	NRITAB=	I	I	I	I	I	I	I		I	I
	[NO]ONEPIUINB		I		I	I	I	I	I	I	I
	[NO]PFXDCEADDR	I	I		I	I	I	I	I	I	I
	PRI|SEC|PEER	I	I	I	I		I	I	I	I	I
	PVCRECONTMR=	I	I			I	I	I	I	I	I
	PVCSETUPTMR=	I	I			I	I	I	I	I	I
	PVCSETUPREJ=	I	I			I	I	I	I	I	I
	[NO]REPDCEADDR	I	I		I	I	I	I	I	I	I
	REQSESSDELAY=		I		I	I	I	I	I	I	I
	[NO]RESETINO		I		I	I	I	I	I	I	I
	[NO]RETPIU	I	I	I	I	I	I	I		I	I
	[NO]RTRBIDREJ		I		I	I	I	I	I	I	I
	[NO]STRIPFAC	I	I		I	I	I	I	I	I	I
	[NO]STRIPRTEIN	I	I		I	I	I	I	I	I	I
	SVCCALLTMR=	I	I			I	I	I	I	I	I
	[NO]TAPWITHCLR	I		I	I	I	I	I	I	I	I
	TCPRBLMT=			I	I	I	I	I	I	I	I
	XID=	I	I	I	I	I	I	I		I	I
PACE=	Supply value for PACE=		I	I	I		I	I	I	I	I
PADPARM=	Supply value for PADPARM=		I			I	I	I	I	I	I
PORT=	Supply value for PORT=	I	I	I	I	I	I	I	I		I
PROTOCOL=	Supply value for PROTOCOL=	I	I	I	I	I	I	I	I		I
SVC0=	Supply value for SVC0=		I		I	I	I	I	I	I	I
SVC3=	Supply value for SVC3=	I	I		I	I	I	I	I	I	I
SVC4=	Supply value for SVC4=		I		I	I	I	I	I	I	I
SVC5=	Supply value for SVC5=		I		I	I	I	I	I	I	I
TAP=	Supply value for TAP=			I	I	I	I	I	I	I	I

PARAMETER	ACTION	XTP	XOT	MCH	MXT	SPU	DFL	DFS	DFX	DMY	SVC
TRAN=	**Supply value for TRAN=**		**I**		**I**	**I**	**I**	**I**	**I**	**I**	**I**
USSTAB=	**Supply value for USSTAB=**		**I**				**I**	**I**	**I**	**I**	**I**

Legend: For the REMOTE *type* column: I=> Invalid, blank=> supported (see Console Command Parameter Action Matrix Keys description on page CONS-21 for additional information).

General notes for MRMT:

1) The modified parameters will take effect for new sessions only.

2) A new CDF can be created from the original CDF plus any changes that are made during HNAS execution using the MLCL and MRMT console commands. The new CDF is produced when the GENNWDF start parameter is specified and the NEWDEFN DD statement is included in the HNAS start JOB. New or modified records are identified in the new CDF by the characters **;NWDF** starting in character position 67. The NEWDEFN file is maintained in memory until HNAS is stopped at which time it is then written to disk. You can display the updated NEWDEFN file in memory by issuing the DNWDF command which is described on page CONS-89.

3) If the MRMT command is going to be used to add an SLU to LUNAME=, SVC0=, SVC4= or SVC5= operand list, a dynamic reconfiguration pool must be specified. This is done using the **OPTIONS=LUDRPOOLCNT=*value*** operand on the BUILD definition statement. You should specify a *value* large enough to accommodate all your SLU addition requests.

MRMT operand syntax:

CUD=NULL|**NONE**|**[*nn*/]*xx...xx*** **(XTP**|**XOT**|**MCH**|**MXT**|**SPU**|**DMY**|**SVC)**
(changed for V2R4M0)

For a **TYPE=XTP|XOT|MCH|SPU|DMY|SVC REMOTE** definition statement, specify **NULL** or **NONE** to delete the existing CUD= operand value.

For a **TYPE=MXT REMOTE** definition statement, specify **NULL** to substitute the MCH CUD= operand value or specify **NONE** to delete the CUD= operand value and not substitute the MCH CUD= operand value. In the latter case, if this MXT is referenced by an SVC0= or SVC5= operand entry on the root MCH, no CUD will be inserted into an outbound Call Request packet even if a CUD= operand value is specified for the MCH.

For a **TYPE=XTP|XOT|MCH|MXT|SPU|DMY|SVC REMOTE** definition statement, specify ***xx...xx*** to add new or replace existing CUD= operand values with the given hexadecimal data starting at the first byte of the CUD= operand (offset 0). You may enter up to 126 paired digits

(***xx***). If you wish to change a portion of an existing CUD= operand, specify an offset value (***nn***) followed by a slash (***/***) before the new CUD= data (***xx...xx***). For example, CUD=4/C1C2 will add or replace CUD= operand data starting at byte 4 with C1C2. If no offset is specified, zero (0) is used. To completely replace an existing CUD= operand, first specify **CUD=NONE** then **CUD=*xx...xx***.

Note: HNAS does not check the CUD= data (***xx...xx***) for standardized X.25 values. The only requirement is that the data must be hexadecimal, the number of digits must not exceed 126 and the digits must be paired (an even number of digits). Although up to 63 bytes of CUD= data are allowed, most routers and PADs will not accept more than 16 bytes when the Protocol ID (CUD byte 0) is set to 01. If more than 16 bytes are specified, the call will most likely be cleared.

DCEADDR=NULL|NONE|*dd...dd* **(XTP|XOT|MCH|MXT|SPU|DMY|SVC)**
(changed for V2R3M0)
(changed for V2R4M0)

For a **TYPE=XTP|XOT|MCH|SPU|DMY|SVC REMOTE** definition statement, specify **NULL** or **NONE** to delete the DCEADDR= operand value. In this case, all non-GATE Call Request packets that HNAS generates **will not include a *calling*** DTE address.

For a **TYPE=MXT REMOTE** definition statement, specify **NULL** to substitute the MCH DCEADDR= operand value or specify **NONE** to delete the DCEADDR= operand value and not substitute the MCH DCEADDR= operand value. In the latter case, if this MXT is referenced by an SVC0= or SVC5= operand entry on the root MCH, no ***calling*** DTE address will be inserted into an outbound Call Request packet even if a DCEADDR= operand value is specified for the MCH.

For a **TYPE=XTP|XOT|MCH|MXT|SPU|DMY|SVC REMOTE** definition statement, specify ***dd...dd*** to replace the DCEADDR= operand value with the given decimal data. In this case, all non-GATE Call Request packets that HNAS generates **will include a *calling*** DTE address. You may enter up to 15 digits (dd).

Note: HNAS does not check the DCEADDR= data (***dd...dd***) for standardized X.25 values. The only requirement is that the data must be decimal and the number of digits must not exceed 15.

DTEADDR=NULL|NONE|*dd...dd* **(XOT|MXT|SPU|DMY|SVC)**
(new for V2R3M0)
(changed for V2R4M0)

For a **TYPE=XOT|MXT|SPU|DMY|SVC REMOTE** definition statement, specify **NULL** or **NONE** to delete the DTEADDR= operand value. In this case, HNAS TAP Call Request packets **will not include a *called*** DTE address.

For a **TYPE=XOT|MXT|SPU|DMY|SVC REMOTE** definition statement, specify ***dd...dd*** to replace the DTEADDR= operand value with the given decimal data. In this case, HNAS TAP Call Request packets **will include a *called*** DTE address. You may enter up to 15 digits (dd).

Note: HNAS does not check the DTEADDR= data (***dd...dd***) for standardized X.25 values. The only requirement is that the data must be decimal and the number of digits must not exceed 15.

FAC=NULL|NONE|[*nn*/]*xx...xx* **(XTP|XOT|MCH|MXT|SPU|DMY|SVC)**
(changed for V2R4M0)

For a **TYPE=XTP|XOT|MCH|SPU|DMY|SVC REMOTE** definition statement, specify **NULL** or **NONE** to delete the FAC= operand value.

For a **TYPE=MXT REMOTE** definition statement, specify **NULL** to substitute the MCH FAC= operand value or specify **NONE** to delete the FAC= operand value and not substitute the MCH FAC= operand value. In the latter case, if this MXT is referenced by an SVC0= or SVC5= operand entry on the root MCH, no facilities will be inserted into an outbound Call Request packet even if a FAC= operand value is specified for the MCH.

For a **TYPE=XTP|XOT|MCH|MXT|SPU|DMY|SVC REMOTE** definition statement, specify ***xx...xx*** to add new or replace existing FAC= operand values with the given hexadecimal data starting at the first byte of the FAC= operand (offset 0). You may enter up to 126 paired digits (***xx***). If you wish to change a portion of an existing FAC= operand, specify an offset value (***nn***) followed by a slash (***/***) before the new FAC= data (***xx...xx***). For example, FAC=6/0101 will add or replace FAC= operand data starting at byte 6 with 0101. If no offset is specified, zero (0) is used. To completely replace an existing FAC= operand, first specify **FAC=NONE** then **FAC=*xx...xx***.

Note: HNAS does not check the FAC= data (***xx...xx***) for standardized X.25 values. The only requirement is that the data must be hexadecimal, the number of digits must not exceed 126 and the digits must be paired (an even number of digits).

IDBLK=*xxx* **(SPU only)**
(new for V2R2M0)

For a **TYPE=SPU REMOTE** definition statement, you may specify up to 3 hex digits (***xxx***) that represents the ID block number for the real remote SPU.

Note: HNAS does not check the IDBLK= parameter data for standardized SDLC network values. The only requirement is that the data must be hexadecimal and the number of digits must not exceed 3.

IDNUM=*xxxxx* **(SPU only)**
(new for V2R2M0)

For a **TYPE=SPU REMOTE** definition statement, you may specify up to 5 hex digits (***xxxxx***) that represents the ID number for the real remote SPU.

Note: HNAS does not check the IDNUM= parameter data for standardized SDLC network values. The only requirement is that the data must be hexadecimal and the number of digits must not exceed 5.

IDLETO=*minutes* **(XTP|MCH|MXT|SPU)**
(new for V2R3M0)

For a **TYPE=XTP|MCH|MXT|SPU REMOTE** definition statement, you may specify an inactivity timeout value from 0 to 255 minutes. A value of 0 inhibits inactivity timeout processing.

INIT={ACTIVE|IDLE} **(XTP|XOT|MCH)**

For a **TYPE=XTP|XOT|MCH REMOTE** definition statement, you may specify the REMOTE initialization options to set or reset. If the REMOTE state changes from ACTIVE to IDLE or vice versa, a VARY RMT ON|OFF command is propagated.

IPADDR=*a.b.c.d*[(*port*)] **(DMY only)**
(new for V2R4M0)

For a **TYPE=DMY REMOTE** definition statement, you may specify an IP address and optional port number in standard dotted notation to be used for the HNAS PING command. The port number can also be changed or set separately (see PORT= parameter below).

LOGTAB=*lgtbname*[-R] **(XTP|MCH|MXT|SPU)**
(apar/changed for V2R4M0)

For a **TYPE=XTP|MCH|MXT|SPU REMOTE** definition statement, specify the name of a valid logon table.

When the -R follower is omitted, the given ***lgtbname*** is compared against all LOGTAB= names currently is use. If the name found, the given LOGTAB= is copied from the *found* REMOTE to the *specified* REMOTE. If the name is not found, the given ***lgtbname*** is LOADed into memory.

When the -R follower is specified, the given ***lgtbname*** is unconditionally loaded into memory even if it is already present. The new version of the table is then propagated to all other REMOTEs that share the table. This means that changes to a table that are made while HNAS is running can be put in to effect without having to stop and restart HNAS.

Note: The MRMT LOGTAB= [-R] support described above was introduced into 240 with APAR 2400032.

LUNAME=*sluname* **(XTP|MCH)**
[-{A|I}]

[/{*pluname*{*}|*}]

(new for V2R4M0)

For the named SLU (***sluname***) in the LUNAME= operand list on a **TYPE=XTP|MCH REMOTE** definition statement, you may change the SLU state (-{**A|I**}), add, replace or delete (*****) a PLU name (***pluname***) or specify the REQSESS indicator (*) following the ***pluname***.

An ***sluname*** value is required for this **fixed form** LUNAME= parameter. If you enter an asterisk (*****) for any LUNAME= entry value (except ***sluname***), the old value is removed.

If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

Examples:

Enter LUNAME=MCH10001/* to remove the PLU association with SLU MCH10001.

Enter LUNAME=MCH10001/CPTCICS* to set the REQSESS indicator for the PLU named CPTCICS that is associated with the SLU named MCH10001.

Enter LUNAME=MCH10001/CPTCICS to reset the REQSESS indicator for the PLU named CPTCICS that is associated with the SLU named MCH10001.

Note: LUNAME= update support for TYPE=XTP|MCH REMOTE was introduced into 240 as Enhancement APAR 2400008.

LUNAME=*sluname* **(DFL only)**
[-{A|I}]
[/{*pluname*|*}]

(new for V2R4M0)

For the named SLU (***sluname***) in the LUNAME= operand list on a **TYPE=DFL REMOTE** definition statement, you may change the SLU state (-{**A|I**}), add, replace or delete (*****) a PLU name (***pluname***).

An ***sluname*** value is required for this **fixed form** LUNAME= parameter. If you enter an asterisk (*****) for any LUNAME= entry value (except ***sluname***), the old value is removed.

If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

Note: LUNAME= update support for TYPE=DFL REMOTE was introduced into 240 as Enhancement APAR 2400028.

LUNAME=*sluname* **(DFS only)**
[-{A|I}]
[/{*pluname*}]
[/{*idnum*|*}]

(new for V2R4M0)

For the named SLU (***sluname***) in the LUNAME= operand list on a **TYPE=DFS REMOTE** definition statement, you may change the SLU state (-{**A|I**}), add or replace but not delete (*****) a PLU name (***pluname***) or add, replace or delete (*) a hex ID number (***idnum***).

An ***sluname*** value is required for this **fixed form** LUNAME= parameter. If you enter an asterisk (*****) for any LUNAME= entry value (except ***sluname*** or ***pluname***), the old value is removed.

If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

Note: LUNAME= update support for TYPE=DFS REMOTE was introduced into 240 as Enhancement APAR 2400028.

LUNAME=*sluname* **(SPU only)**
[-{A|I}]
[/{*rpcecnt*|*}]
[/{*spcecnt*|*}]
[/{*applid*|*}]
[/{*mxtname*|*}]

(new for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

For the named SLU (***sluname***) in the LUNAME= operand list on a **TYPE=SPU REMOTE** definition statement, you may change the SLU state (-{**A|I**}), add, replace or delete (*****) a receive pacing count (***rpcecnt***), a send pacing count (***spcecnt***), an APPLNAME= operand index (***applid***) and/or the name of the TYPE=MXT REMOTE associated with the LUNAME= operand entry (***mxtname***).

An ***sluname*** value is required for this **fixed form** LUNAME= parameter. If you enter an asterisk (*****) for any LUNAME= entry value (except ***sluname***), the old value is removed.

If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

Note: When no receive or send pacing count is associated with an ***sluname*** in the LUNAME= operand, the corresponding value from the PACE= operand on the same TYPE=SPU REMOTE definition statement is used.

```
LUNAME=(locaddr,                                              (SPU only)
        {sluname|*}
         [-{A|I}]
         [/{rpcecnt|*}]
         [/{spcecnt|*}]
         [/{applid|*}]
         [/{mxtname|*}],...
         :
        {sluname|*}
         [-{A|I}]
         [/{rpcecnt|*}]
         [/{spcecnt|*}]
         [/{applid|*}]
         [/{mxtname|*}])
                                                   (new for V2R3M0)
                                                (changed for V2R4M0)
```

Starting at the local address identified by the ***locaddr*** value in the LUNAME= operand list on a **TYPE=SPU REMOTE** definition statement, you may add, replace or delete (*****) an SLU name (***sluname***) change the SLU state (-{**A|I**}), add, replace or delete (*****) a receive pacing count (***rpcecnt***), a send pacing count (***spcecnt***), an APPLNAME= operand index (***applid***) and/or the name of the TYPE=MXT REMOTE associated with the LUNAME= operand entry (***mxtname***).

A ***locaddr*** value is required for this **list form** LUNAME=(*list*) parameter. If you enter an asterisk (*****) for any LUNAME= entry value (except ***sluname***), the old value is removed. If you enter an asterisk (*****) for the ***sluname*** value, the entire LUNAME= operand entry is removed.

If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

Note: When no receive or send pacing count is associated with an ***sluname*** in the LUNAME= operand, the corresponding value from the PACE= operand on the same TYPE=SPU REMOTE definition statement is used.

To summarize, for a **TYPE=SPU REMOTE** definition statement, you may add, replace or delete **multiple** LUNAME= operand list entries using the **list form** of the LUNAME= parameter where:

locaddr is the starting LOCADDR entry in the LUNAME= operand.
sluname is an existing or new SLU name for the selected entry.
{A|I} is an existing or new SLU state for the selected entry.
rpcecnt is a changed or new receive pacing count for the selected entry.
spcecnt is a changed or new send pacing count for the selected entry.
applid is a changed or new APPLNAME= operand index for the selected entry.
mxtname is a changed or new TYPE=MXT REMOTE name for the selected entry.

***** indicates that the selected value is to be removed from LUNAME= operand.

Note: If ***locaddr*** is greater than the existing LUNAME= operand list size, LUs will be allocated from the LU Dynamic Reconfiguration Pool (**LUDRPOOLCNT=** suboperand of the OPTIONS= operand on BUILD definition statement) to satisfy the addition of new list entries up to a maximum number of 255. If an LUNAME= operand list entry is being deleted, the associated LU is returned to the LUDRPOOL for later use. The LU name for an allocated LU is set to the name you specify in the MRMT console command.

Note: If ***sluname*** represents a new name for an existing or new LUNAME= operand entry, it must also be added and activated in an HNAS Application Major Node File (AMNF).

Note: If an LU is active at a ***locaddr*** entry, it will be deactivated (UNBOUND, ACB closed) if its ***sluname*** is being changed or deleted.

MAXDATA=*bytecnt* **(SPU only)**
(new for V2R2M0)

For a **TYPE=SPU REMOTE** definition statement, you may specify a ***bytecnt*** value between 64 and 8192. The ***bytecnt*** value represents the maximum number of bytes that can be delivered to the real remote SPU in a single PIU. This value account for the RU, RH and TH (6 bytes for a PUT2 and 2 bytes for a PUT1).

OPTIONS=[,CLOTINITYP={NONE|BIND|TIMER|CONSOLE}]	**SPU**
[,CLOTFAILRTYLMT=*count*]	**SPU**
[,CLOTCONLMT=*count*]	**SPU**
[,{CLOSEONTAPFAILURE|NOCLOSEONTAPFAILURE}]	**XTP|XOT**
[,{CUD0SELECTSLU|NOCUD0SELECTSLU}]	**XTP|MCH|MXT**
[,{DATAF|NODATAF}]	**DFX**
[,{DATAFAM|NODATAFAM}]	**DFX**
[,{DELAYBINDRESP|NODELAYBINDRESP}]	**XTP|MCH**
[,{ECHODTEADDR|NOECHODTEADDR}]	**MCH**
[,{ECHOFAC|NOECHOFAC}]	**MCH**
[,{EMSGE|NOEMSGE}]	**DFX**
[,{IDTST|NOIDTST}]	**DFX**
[,{IMS|NOIMS}]	**DFX**
[,{INHIBITBIDREJ|NOINHIBITBIDREJ}]	**XTP|MCH**
[,{LLC0CTCPCHK|NOLLC0CTCPCHK}]	**MCH**
[,{LLC5CTCPCHK|NOLLC5CTCPCHK}]	**MCH**
[,MCHTMR=*seconds*]	**XTP|MCH**
[,NRITAB=*nrtbname*[-R]]	**DFX**
[,{ONEPIUINB|NOONEPIUINB}]	**XTP|MCH**
[,{PEER|PRI|SEC}]	**SPU**
[,{PFXDCEADDR|NOPFXDCEADDR}]	**MCH**
[,PVCRECONTMR=*seconds*]	**MCH|MXT**
[,PVCSETUPTMR=*seconds*]	**MCH|MXT**
[,PVCSETUPREJ=*code*]	**MCH|MXT**
[,{REPDCEADDR|NOREPDCEADDR}]	**MCH**

```
[,REQSESSDELAY=seconds]                                          XTP|MCH
[,{RESETINO|NORESETINO}]                                         XTP|MCH
[,{RETPIU|NORETPIU}]                                                 DFX
[,{RTRBIDREJ|NORTRBIDREJ}]                                       XTP|MCH
[,{STRIPFAC|NOSTRIPFAC}]                                             MCH
[,{STRIPRTEIN|NOSTRIPRTEIN}]                                         MCH
[,SVCCALLTMR=seconds]                                            MCH|MXT
[,{TAPWITHCLR|NOTAPWITHCLR}]                                         XOT
[,TCPRBLMT=bfrcnt]                                               XTP|XOT
[,XID={NO|STD|TAB|TABSTD|(idnum,idcnt)}]                             DFX
                                                 (changed for V2R2M0)
                                                 (changed for V2R3M0)
                                                 (changed for V2R4M0)
```

Specify the REMOTE options to set or reset.

For a **TYPE=XTP|MCH|MXT REMOTE** definition statement, you may specify OPTIONS= operand values of **[NO]CUD0SELECTSLU**.

Note: The **[NO]CUD0SELECTSLU** option was introduced into 240 with enhancement APAR 2400074.

For a **TYPE=XTP|MCH REMOTE** definition statement, you may specify OPTIONS= operand values of **[NO]DELAYBINDRESP**, **[NO]INHIBITBIDREJ**, **MCHTMR=*seconds*** (where ***seconds***=4-60), **[NO]ONEPIUINB**, **REQSESSDELAY=*seconds*** (where ***seconds***=1-254), **[NO]RESETINO** and/or **[NO]RTRBIDREJ**.

For a **TYPE=MCH|MXT REMOTE** definition statement, you may specify OPTIONS= operand values of **PVCRECONTMR=*seconds*** (where ***seconds***=5-254), **PVCSETUPTMR=*seconds*** (where ***seconds***=10-254), **PVCSETUPREJ=*code*** (where ***code***=01-7F except for 12) and/or **SVCCALLTMR=*seconds*** (where ***seconds***=10-254).

Note: T21= can be specified as an alternate name for the **SVCCALLTMR=** suboperand.

For a **TYPE=MCH REMOTE** definition statement only, you may specify OPTIONS= operand values of **[NO]ECHODTEADDR**, **[NO]ECHOFAC**, **[NO]LLC0CTCPCHK**, **[NO]LLC5CTCPCHK**, **[NO]PFXDCEADDR**, **[NO]REPDCEADDR**, **[NO]STRIPFAC** and/or **[NO]STRIPRTEIN**.

For a **TYPE=XTP|XOT REMOTE** definition statement, you may specify OPTIONS= operand values of **[NO]CLOSEONTAPFAILURE** and **TCPRBLMT=*bfrcnt*** (where ***bfrcnt***=1-7).

Note: The **[NO]CLOSEONTAPFAILURE** option was introduced into 240 with enhancement APAR 2400055.

For a **TYPE=XOT REMOTE** definition statement only, you may specify OPTIONS= operand values of **[NO]TAPWITHCLR**.

Note: For HNAS release V2R2M0, Keep Alive logic used a Call Request packet as the shoulder TAP request packet. Some router configurations cause the this Call Request packet to

be propagated to the connected X.25 network which is an undesirable side effect. Normally, the router simply 'eats' the HNAS Call Request and returns a Clear Request which satisfies the HNAS TAP requirement. In order to eliminate this side effect and still permit the HNAS TAP logic to function, the TAPWITHCLR option has been added which will condition HNAS to use a Clear Request rather than a Call Request as the TAP request packet.

Note: For HNAS releases before V2R2M0, a Clear Request was used as the TAP request but this was changed to a Call Request because the IOS for some Cisco routers do not respond to Clear Requests. This can make an otherwise functioning router appear down. This is why the TAP request was changed to a Call Request which ensures a response from the router when it and it's XOT component are both active. For installations that prefer, and can use a Clear Request as the TAP request, the TAPWITHCLR option may specified.

For a **TYPE=SPU REMOTE** definition statement only, you may specify OPTIONS= operand values of **PEER**, **PRI, SEC, CLOTINITYP={NONE|BIND|TIMER|CONSOLE}, CLOTFAILR-TYLMT=*count*** (where ***count***=0-254) and/or **CLOTCONLMT=*count*** (where ***count***=0-254).

For a **TYPE=DFX REMOTE** definition statement only, you may specify OPTIONS= operand values of **[NO]DATAF**, **[NO]DATAFAM**, **[NO]EMSGE**, **[NO]IDTST**, **[NO]IMS**, **NRI-TAB=*nrtbname*[-R]**, **[NO]RETPIU** and/or **XID={NO|STD|TAB|TABSTD|(*idnum*,*idcnt*)}**.

Note: The **[NO]IDTST** option was introduced into 240 with enhancement APAR 2400074.

For NRITAB=:

When the -R follower is omitted, the given ***nrtbname*** is compared against all NRITAB= names currently is use. If the name found, the given NRITAB= is copied from the *found* REMOTE to the *specified* REMOTE. If the name is not found, the given ***nrtbname*** is LOADed into memory.

When the -R follower is specified, the given ***nrtbname*** is unconditionally loaded into memory even if it is already present. The new version of the table is then propagated to all other REMOTEs that share the table. This means that changes to a table that are made while HNAS is running can be put in to effect without having to stop and restart HNAS.

Note: The MRMT NRITAB= [-R] support described above was introduced into 240 with APAR 2400032.

```
PACE={NONE|NULL|pktcnt}|                                    (XTP)
      ({rpcecnt|*},{spcecnt|*})                             (SPU)
                                                  (new for V2R2M0)
```

For a **TYPE=XTP REMOTE** definition statement, specify **NULL|NONE** to delete the PACE= operand value.

For a **TYPE=SPU REMOTE** definition statement, specify **NULL** to substitute the pacing values in the BIND image or specify **NONE** to delete the PACE= operand value and not substi-

tute the pacing values from the BIND image. In the latter case, no inbound or outbound pacing will be performed for all SLUs on this SPU.

For a **TYPE=XTP REMOTE** definition statement, specify a decimal ***pktcnt*** value between 0 and 254.

For a **TYPE=SPU REMOTE** definition statement, you may add, replace or delete (*****) a receive pacing count (***rpcecnt***) and/or a send pacing count (***spcecnt***). The values are entered as a sublist enclosed in parentheses. If you enter an asterisk (*****) for either pacing count, the old value is removed. When no receive or send pacing count is associated with the PACE= operand value, the corresponding value from the BIND image for each SLU on the SPU is used.

PADPARM=NULL|NONE|(*pn*1,*pv*1,...,*pn*k,*pv*k) **(XTP|MCH|MXT)**

For a **TYPE=XTP|MCH REMOTE** definition statement, specify **NULL|NONE** to delete the PADPARM= operand value.

For a **TYPE=MXT REMOTE** definition statement, specify **NULL** to substitute the MCH PADPARM= operand value or specify **NONE** to delete the PADPARM= operand value and not substitute the MCH PADPARM= operand value. In the latter case, if this MXT is referenced by an SVC0= or SVC5= operand entry on the root MCH, no PAD parameters will be transmitted to the remote DTE even if a PADPARM= operand value is specified for the MCH.

For a **TYPE=XTP|MCH|MXT REMOTE** definition statement, specify **(*pn*1,*pv*1,...,*pn*k,*pv*k)** to replace or modify the PADPARM= operand value with the given PAD parameter data. You may enter up to 127 decimal number/value pairs (***pn*i**,***pv*i**) *enclosed in parenthesis*. If a PAD parameter number already exists, the new value will replace the old value. If a PAD parameter number is new, the number and its value will be added to the PAD parameter list.

Note: HNAS does not check the PADPARM= data **(*pn*1,*pv*1,...,*pn*k,*pv*k)** for standardized X.25 values. The only requirement is that the data must be decimal in the range 0-255 and the number of number/value pairs must not exceed 127. Note that all PADPARM= values are treated as international PAD parameters.

PORT=*number* **(DMY only)**
(new for V2R4M0)

For a **TYPE=DMY REMOTE** definition statement, you may specify a port number to be used for the HNAS PING command. The port number can also be specified as part of the IP address (see IPADDR= parameter above).

PROTOCOL={XTP|XOT} **(DMY only)**
(new for V2R4M0)

For a **TYPE=DMY REMOTE** definition statement, you may specify the X.25 encapsulation protocol to be used for the HNAS PING command.

```
SVC0=sluname=[{newname|*}]                                          (XTP|MCH)
     [+gluname][-{A|I}]
     [/{dteaddr1|Xidnum1|<rmtname1>|*}
      [-{dteaddr2|Xidnum2|<rmtname2>|*}
        [-{dteaddr3|Xidnum3|<rmtname3>|*}]]{T|O|I}[{applid|*}]]
     [/{mxtname|*}]
     [/{cud|*}]
                                                    (changed for V2R2M0)
                                                    (changed for V2R4M0)
```

For the named SLU (***sluname***) in the SVC0= operand list on a **TYPE=XTP|MCH REMOTE** definition statement, you may add, delete or modify the following SLU entry values:

***sluname*=[{*newname*|*}]**

If *sluname* is specified by itself and *sluname* exists in the SVC0= operand list, the SLU remains unchanged but suboperands associated with the SLU may be modified.

If *sluname*=*sluname* is specified and *sluname* does not exist, the new SLU is added. For SLU add requests, the same name must be specified on both sides of the equal sign.

If *sluname*=*newname* is specified and *newname* is unique, the SLU is renamed. If the SLU is active at the time the rename takes place, the call is cleared and the ACB is closed. For this reason, you should verify the SLU state using the DLU LUNM=*sluname* command before making this change.

If *sluname*=* is specified, the entire SLU entry is removed from the SVC0= operand. If the SLU is active at the time the deletion takes place, the call is cleared and the ACB is closed. For this reason, you should verify the SLU state using the DLU LUNM=*sluname* command before making this change.

If the *sluname* is not connected but its ACB is opened when a name change or deletion is requested (for example, awaiting a callout BIND), the following error message is generated:

```
NASC421E SLU sluname IS NOT IDLE, REQUIRED
```

In this case, you must issue V NET,INACT,ID=*sluname* at the VTAM console to deactivate the SLU.

If the *newname* is already defined when a name change is requested, the following error message is generated:

```
NASC422E SLU newname ALREADY EXISTS, REQUIRED
```

If the *newname* is unique when an addition is requested but no dynamic LU pool is defined or all dynamic LUs are in use, the following error message is generated:

```
NASC423E SLU sluname COULD NOT BE ALLOCATED, REQUIRED
```

In this case, you must ensure the value specified for OPTIONS=LUDRPOOLCNT=*value* on the BUILD definition statement is large enough to accommodate all your dynamic LU addition requests.

+gluname

is a generic SLU name that will be used in place of the real SLU name when a SETLOGON request is presented to VTAM during *callin* processing.

Note: *sluname+gluname* support was introduced into 240 as Enhancement APAR 2400084.

-{A|I}

Is the SLU state. If omitted, no change is effected. If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

{*dteaddr*1|X*idnum*1|<*rmtname*1>|*}
[-{*dteaddr*2|X*idnum*2|<*rmtname*1>|*}
[-{*dteaddr*3|X*idnum*3|<*rmtname*1>|*}]]

You may specify up to three (3) ***dteid*i** values (***dteid*i** is an abbreviation for ***dteaddr*i**, **X*idnum*i** or **<*rmtname*i>**), each of which must be separated by a hyphen (-). Each ***dteaddr*i** can be up to fifteen (15) decimal digits in length, each **X*idnum*i** can be up to fourteen (14) paired hex digits in length and each **<*rmtname*i>** must identify a TYPE=MXT|SVC REMOTE definition statement whose name must be enclosed in non-equivalence symbols (<>). The connection identifier follows the ***dteid*i** list.

Note: When the first character of the ***dteaddr*i** is an **X**, it indicates that a hex **X*idnum*i** value follows rather than a decimal ***dteaddr*i** value. You should always code an even number of hex digits using zero (0) as the right most pad digit if the **X*idnum*i** value is odd.

If an **<*rmtname*i>** value is specified, it must point at a TYPE=MXT|SVC REMOTE definition statement. The DTEADDR= operand on the named MXT|SVC will provide the ***dteaddr*i** value for inbound ***calling*** DTE address matching and for outbound ***called*** DTE address targetting. Additionally, for outbound calls, the DCEADDR=, FAC= and CUD= operands on the named MXT|SVC provide values for the ***calling*** DTE address, facilities data and call user data, respectively.

If an asterisk (*) is specified for a ***dteid*i** value, the value is removed from the SLU entry.

When a hyphen is entered without a preceding ***dteid*i** value, it acts as a place holder preserving the value of the skipped *dteidi* list entry.

Note: <*rmtnamei*> support was introduced into 240 as Enhancement APAR 2400003.

{I|O|T} (connection identifier)

Specifies that the SLU is used for callin (I), callout (O) or twoway (T) communication. The connection identifier may be changed dynamically.

{*applid*|*}

Specifies an APPLNAME= operand index value that follows the connection identifier. An *applid* value is permitted for callin and twoway SLUs only (type of I|T). If an asterisk (*) is specified for an *applid* value, the value is removed from the SLU entry.

{*mxtname*|*}

Specifies the name of a TYPE=MXT REMOTE definition statement that is to be associated with the SLU. The *mxtname* is valid for all SLU types. If an asterisk (*) is specified for an *mxtname* value, the value is removed from the SLU entry.

{*cud*|*}

Specifies 4-bytes (8 hex digits) of call user data for callout or twoway SLUs. If an asterisk (*) is specified for a *cud* value, the value is removed from the SLU entry. If you require more than 4-bytes of CUD, you should associate an MXT with the SLU and use the CUD= operand on that MXT. The CUD= operand allows for up to 63-bytes of call user data.

Note: If the connection identifier is omitted, the existing connection identifier remains unchanged. If this is a new SVC0= SLU entry, (**I**) is assumed.

Examples:

1) Change/add a DTE address for the existing SLU named MCH10001 on MCH1 then display the result.

```
RNM=MCH1 MRMT SVC0=MCH10001/123456, DRMT SVC0=

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0001
              MCH10001/123456I/
```

2) Associate the MXT named MXT1 with the existing SLU named MCH10001 on MCH1 then display the result.

```
RNM=MCH1 MRMT SVC0=MCH10001//MXT1, DRMT SVC0=

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0001
              MCH10001/123456I/MXT1
```

3) Add an X*idnum* and <*rmtname*> value to the *dteid*i list while leaving the *dteaddr* value alone:

```
MRMT MCH1 SVC0=MCH10001/-X6789-<MTX0>, DRMT MCH1 SVC0

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0020
              MCH10001/123456-X6789-<MXT0>I/MXT1
```

Note: When a hyphen is entered without a preceding value, it acts as a place holder preserving the value of the skipped *dteid*i list entry (123456 in this case).

4) Change the connection identifier from callin to twoway and add an *applid* of 2 for the existing SLU named MCH10001 on MCH1 then display the result.

```
RNM=MCH1 MRMT SVC0=MCH10001/T2, DRMT SVC0=

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0001
              MCH10001/123456-X6789-<MXT0>T02/MXT1
```

5) Add a new SLU named MCH1NEW on MCH1 then display the result.

```
RNM=MCH1 MRMT SVC0=MCH1NEW=MCH1NEW/45456677T2//C0000000, DRMT SVC0=

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0002
              MCH10001/123456-X6789-<MXT0>T02/MXT1
              MCH1NEW /45456677T02//C0000000
```

6) Change the name of the existing SLU named MCH1NEW on MCH1 to MCH10002 then display the result.

```
RNM=MCH1 MRMT SVC0=MCH1NEW=MCH10002, DRMT SVC0=

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0002
              MCH10001/123456-X6789-<MXT0>T02/MXT1
              MCH10002/45456677T02//C0000000
```

7) Delete the existing SLU named MCH10002 on MCH1 then display the result.

```
RNM=MCH1 MRMT SVC0=MCH10002=*, DRMT SVC0=

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0001
              MCH10001/123456-X6789-<MXT0>T02/MXT1
```

SVC3=*spuname* **(MCH only)**

[/{*dteaddr*|*}[{I|O}]]
[/{*mxtname*|*}]

(new for V2R2M0)

For the named SPU (***spuname***) in the SVC3= operand list on a **TYPE=MCH REMOTE** definition statement that identifies a callin resource (**I**) or callout resource (**O**), you may add, replace or delete (*****) the value for the resource identifier with a decimal DTE address value **(*dteaddr***). You may enter up to 15 decimal digits for ***dteaddr***. If you enter an asterisk (*****) for the resource identifier, the old value is removed. You may also add, change or delete (*) the name of the TYPE=MXT REMOTE (***mxtname***) associated with the SVC3= operand entry.

Note: If the callin (**I**) or callout (**O**) delimiter is entered, it must match the value that was specified for the SVC3= operand entry in the CDF. If the delimiter is omitted, callin (**I**) is assumed.

SVC4=*sluname*=[{*newname*|*}] **(XTP|MCH)**
[-{A|I}]

(changed for V2R3M0)
(changed for V2R4M0)

For the named SLU (***sluname***) in the SVC4= operand list on a **TYPE=XTP|MCH REMOTE** definition statement, you may add, delete or modify the following SLU entry values:

***sluname*=[{*newname*|*}]**

If *sluname* is specified by itself and *sluname* exists in the SVC4= operand list, the SLU remains unchanged.

If *sluname=sluname* is specified and *sluname* does not exist, the new SLU is added. For SLU add requests, the same name must be specified on both sides of the equal sign.

If *sluname=newname* is specified and *newname* is unique, the SLU is renamed. If the SLU is active at the time the rename takes place, the call is cleared and the ACB is closed. For this reason, you should verify the SLU state using the DLU LUNM=*sluname* command before making this change.

If *sluname*=* is specified, the entire SLU entry is removed from the SVC4= operand. If the SLU is active at the time the deletion takes place, the call is cleared and the ACB is closed. For this reason, you should verify the SLU state using the DLU LUNM=*sluname* command before making this change.

If the *sluname* is not connected but its ACB is opened when a name change or deletion is requested (for example, awaiting a callout BIND), the following error message is generated:

```
NASC421E SLU sluname IS NOT IDLE, REQUIRED
```

In this case, you must issue V NET,INACT,ID=*sluname* at the VTAM console to deactivate the SLU.

If the *newname* is already defined when a name change is requested, the following error message is generated:

```
NASC422E SLU newname ALREADY EXISTS, REQUIRED
```

If the *newname* is unique when an addition is requested but no dynamic LU pool is defined or all dynamic LUs are in use, the following error message is generated:

```
NASC423E SLU sluname COULD NOT BE ALLOCATED, REQUIRED
```

In this case, you must ensure the value specified for OPTIONS=LUDRPOOLCNT=*value* on the BUILD definition statement is large enough to accommodate all your dynamic LU addition requests.

-{A|I}

Is the SLU state. If omitted, no change is effected. If ***sluname*-{A|I}** is entered and this changes the current SLU state, an appropriate **VARY *sluname* ON|OFF** console command is issued internally.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

```
SVC5=sluname=[{newname|*}]                                    (XTP|MCH)
     [+gluname][-{A|I}]
     [/{dteaddr1|Xidnum1|<rmtname1>|*}
      [-{dteaddr2|Xidnum2|<rmtname2>|*}
       [-{dteaddr3|Xidnum3|<rmtname3>|*}]]{T|O|I}[{applid|*}]]
     [/{mxtname|*}]
     [/{cud|*}]
                                                  (changed for V2R2M0)
                                                  (changed for V2R4M0)
```

Note: The syntax for the SVC5= parameter is identical to that for the SVC0= operand. Please refer to the description of SVC0= on page CONS-194 for details.

```
TAP=value                                                     (XTP|XOT)
```

For a **TYPE=XTP|XOT REMOTE** definition statement, specify a shoulder tap timeout value between 0 and 4095. If 0 is specified, 'Keep Alive' processing is inhibited for the associated router. Please refer to Chapter 4 description of TAP= operand for usage restrictions.

```
TRAN={NO|USER|EVEN|ODD|MARK|SPACE|                            (XTP|MCH)
      NPSIEVEN|NPSIODD|NPSIMARK|NPSISPACE}
                                                  (changed for V2R3M0)
```

For a **TYPE=XTP|MCH REMOTE** definition statement, specify a valid translate table type.

USSTAB=*ustbname*[-R] **(XTP|MCH|MXT|SPU)**
(apar/changed for V2R4M0)

For a **TYPE=XTP|MCH|MXT|SPU REMOTE** definition statement, specify the name of a valid USS table.

When the -R follower is omitted, the given ***ustbname*** is compared against all USSTAB= names currently is use. If the name found, the given USSTAB= is copied from the *found* REMOTE to the *specified* REMOTE. If the name is not found, the given ***ustbname*** is LOADed into memory.

When the -R follower is specified, the given ***ustbname*** is unconditionally loaded into memory even if it is already present. The new version of the table is then propagated to all other REMOTEs that share the table. This means that changes to a table that are made while HNAS is running can be put in to effect without having to stop and restart HNAS.

Note: The MRMT USSTAB= [-R] support described above was introduced into 240 with APAR 2400032.

General Notes for APAR 2400032:

1) Because the transmission of a USSMSG in a USSTAB can span task dispatches (and TCPIP interrupts), HNAS will refuse to reload a USSTAB if it is currently being used for a USSMSG transmission. In this case, the following message is issued:

```
NASC031W USSTAB ISTINCDT IS CURRENTLY IN USE, RETRY MRMT COMMAND LATER
```

This is the case because of a potential ABEND that can result due to an addressing exception if the new table alters USSMSGs currently being transmitted.

Further, if you attempt to load a table that is not appropriate for the specified operand, for example a USS table is specified for a LOGON table (e.g., LOGTAB=ISTINCDT), the following message is issued:

```
NASC030E LOGTAB ISTINCDT HAS INVALID FORMAT, MRMT COMMAND ABORTED
```

2) The following message is also issued if you specify the wrong table in the CDF, for example a USS table is specified for a LOGON table (e.g., LOGTAB=ISTINCDT):

NAS1311E REMOTE *mchname* LOGTAB=ISTINCDT IS NOT A VALID LOGON TABLE, REQUIRED

*** THIS NEW MESSAGE WILL PREVENT HNAS FROM STARTING ***

Prior to APAR 2400032, you would not have known that a bad table was specified until it was accessed during call setup.

For a bad USSTAB=, the following message is issued and the call is cleared:

```
NAS5702E LU sluname ADDRESSES INV USSTAB
```

For a bad LOGTAB= value, no action is taken (treated as if no LOGTAB= value was specified). In this case USSTAB= is used. If the USSTAB= operand is (also) missing, the following message is issued and the call is cleared:

```
NAS5703E LU sluname MISSING USSTAB
```

Note: If invalid data is entered for an operand, its value is not changed and the MRMT command is aborted at that point.

MRMT Examples:

As an illustration of how the MRMT and DRMT commands can be used together, consider the following examples. Note that you should always issue a DRMT command before making any changes so you will have a log of initial values.

Initially, the following display is produced when **RNM=MCH1 DRMT SVC0** is entered (RNM=MCH1 sets the REMOTE name for subsequent MRMT and DRMT commands).

```
RNM=MCH1 DRMT SVC0

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0005
              MCH10001/I/
              MCH10002/I/
              MCH10003/66660101O/
              MCH10004/666666666666666O/
              MCH10005/666612345678901O/MXT1
```

1) Change the DTE address for the SLU named MCH10004 from 666666666666666 to 3036102400 then verify the change.

```
MRMT SVC0=MCH10004/3036102400
DRMT SVC0

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0005
              MCH10001/I/
              MCH10002/I/
              MCH10003/66660101O/
              MCH10004/30361024000O/
              MCH10005/666612345678901O/MXT1
```

2) Define a TYPE=MXT REMOTE named MXT1 for the SLU named MCH10004 then verify the change.

```
MRMT SVC0=MCH10004//MXT1
DRMT SVC0

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0005
              MCH10001/I/
              MCH10002/I/
              MCH10003/66660101O/
              MCH10004/30361024000O/MXT1
              MCH10005/666612345678901O/MXT1
```

3) Supply a TYPE=MXT REMOTE named MXT1 and change the callout DTE address for the SLU named MCH10003 from 66660101 to 3036102500 then verify the change.

```
MRMT SVC0=MCH10003/30361025000/MXT1
DRMT SVC0

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0005
              MCH10001/I/
              MCH10002/I/
              MCH10003/30361025000/MXT1
              MCH10004/30361024000/MXT1
              MCH10005/666612345678901O/MXT1
```

4) Add a TYPE=MXT REMOTE named MXT1 and a callin DTE address of 3036102600 for the SLU named MCH10002 then verify the change.

```
MRMT SVC0=MCH10002/3036102600/MXT1
DRMT SVC0

RMTNAME  OPERAND  TYPE
MCH1     00074920 MCH
         SVC0=0005
              MCH10001/I/
              MCH10002/3036102600I/MXT1
              MCH10003/30361025000/MXT1
              MCH10004/30361024000/MXT1
              MCH10005/666612345678901O/MXT1
```

PAUSE Command - Pause Console Processing (Sleep)

(new for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	PAUSE	***seconds***	**N**	**240**

This command is used to suspend or delay execution of console commands that follow the PAUSE command for the amount of time specified by the *seconds* value. The PAUSE command is useful for delaying queued command execution (CONCMDQ= operand on the BUILD definition statement) when HNAS is first started until initialization activity has subsided or to delay commands provided in a command list (EXEC *ddname*|(*cmd*1,...,*cmd*n).

You may specify a ***seconds*** value between 1 and 3600 (1-hour). If ***seconds*** is omitted, 10 will be used.

PFXWTO Command - WTO Prefix Control (Privileged)

(changed for V2R3M0)

Entered from the local console

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	PFXWTO	**[{ON|NASNAME|*text*|OFF}]** **[CONS {ON|OFF}]**	**A** **A**	**230** **230**

Entered from a remote console

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	PFXWTO	**[{ON|TIME|OFF}]**		

For local consoles, this command is used to enable (**ON**|**NASNAME**) or disable (**OFF**) WTO prefixing for alarm messages or to enable (**CONS ON**) or disable (**CONS OFF**) WTO prefixing for console command output using the text supplied by the NASNAME= operand of the BUILD definition statement. If the *text* parameter is specified, WTO prefixing is enabled and the *text* value will be used to override the NASNAME= operand value. From 1 to 8 non-blank characters may be specified.

WTO prefix support is provided so that data center operations personnel can quickly identify HNAS generated messages in the local console (**SYSCONS**) log.

For remote consoles, this command is used to enable (**ON**|**TIME**) or disable (**OFF**) WTO prefixing using a time stamp.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

PFXWTO Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER	LCL	RMT
ON	**OFF**	**1**		
OFF	**ON**	**2**		
NASNAME	***text***	**3**		**I**
text	**NASNAME**	**4**		**I**
CONS ON	**CONS OFF**	**5**		**I**
CONS OFF	**CONS ON**	**6**		**I**
TIME	**OFF**	**1**	**I**	

Console Commands

- Enter **PFXWTO [ON|NASNAME]** from a local console to cause the text provided by the NAS-NAME= operand of the BUILD definition statement to be appended to the beginning of each **alarm** WTO written to the **SYSCONS**. This processing does not affect local console command output.

- Enter **PFXWTO OFF** from a local console to inhibit WTO prefix support for alarm message output.

- Enter **PFXWTO CONS [ON]** from a local console to cause the text provided by the NAS-NAME= operand of the BUILD definition statement to be appended to the beginning of each **console command** WTO written to the **SYSCONS**. This processing requires that the **SHOW CONS ON** option must also be in effect. This processing does not affect alarm output.

- Enter **PFXWTO CONS OFF** from a local console to inhibit WTO prefix support for console command output.

- Enter **PFXWTO *text*** from a local console to cause the *text* value to be appended to the beginning of each **alarm** WTO in lieu of the NASNAME= operand value. Up to 8 non-blank characters may be specified.

- Enter **PFXWTO *text* PFXWTO OFF CONS ON** if you only want console output to be prefixed with the *text* value.

- Enter **PFXWTO [ON|TIME]** from a remote console to cause the text provided by the time of day clock to be appended to the beginning of each WTO written to the remote console.

- Enter **PFXWTO OFF** to inhibit WTO prefix support.

General notes for PFXWTO:

1) HNAS activates with **PFXWTO OFF** set in the absence of the **PFXWTO** start parameter.

2) The **PFXWTO [ON]**, **PFXWTO OFF** and **PFXWTO *text*** start parameters perform the same function as the **PFXWTO [ON]**, **PFXWTO OFF** and **PFXWTO *text*** console commands, respectively, for *global* message prefix control. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

PING Command - Ping Remote Router and/or MCH Link (Privileged)

(new for V2R3M0)
(changed for V2R4M0)

PINGing based on supplied parameters

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[IPADDR=*aaa.bbb.ccc.ddd*[(*ppppp*)]]	PING	[XOT|XTP]	N	230
[CLDADDR=*ddd...ddd*]		[*aaa.bbb.ccc.ddd*[(*ppppp*)]		
[CLGADDR=*ddd...ddd*]		[*ddd...ddd*]		
[IFN=*ifn*]		[*ifn*]		
[VCN=*vcn*]		[*vcn*]		

or PINGing based on REMOTE parameters with optional overrides

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*dmyname*]	PING	[*dmyname*]	N	240
[IFN=*ifn*]		[XOT|XTP]		
[VCN=*vcn*]		[*aaa.bbb.ccc.ddd*[(*ppppp*)]		
		[*ddd...ddd*]		
		[*ifn*]		
		[*vcn*]		

This command will transmit a Call Request packet to the remote DTE identified by the command modifiers (default values) or the specified **positional** command parameters.

The purpose of this command is to verify connectivity from HNAS to the destination router and beyond (including the X.25 serial interface and network when configured with the appropriate X.25 DTE|DCE addresses, routes, etc.).

PING Call Request Parameter Hierarchy Matrix

PARAMETER	When DMY specified	When DMY not specified
called DTE address	DTEADDR= from DMY, PING argument override or null	CLDADDR= modifier, PING argument override or null
calling DTE address	DCEADDR= from DMY or null	CLGADDR= modifier or null
facilities data	FAC= from DMY or null	defaults to 080100420707430202
call user data	CUD= from DMY or null	defaults to 01000000HNASPING

Note: If the operand of the hierarchy list for a particular Call Request parameter is undefined (e.g., NONE|NULL), the corresponding Call Request field will be null.

If you supply the name of a TYPE=DMY REMOTE definition statement via the RNM= modifier, it will be used to supply the IPADDR=, PORT=, PROTOCOL= (either XTP or XOT), CUD=, DTEADDR= (same function as CLDADDR=), DCEADDR= (same function as CLGADDR=) and FAC= values for the PING command. If you wish to override certain DMY operands, you can enter the associated override values as positional PING command arguments. If the PING is for an XTP call, the *ifn* and *vcn* values must be supplied via the respective command modifiers or as positional parameters. In the latter case, commas should be used to identify omitted positional parameters (,,) that precede *ifn* and *vcn*. The use of RNM=*dmyname* provides more flexibility than simply using the other command modifiers since **all** Call Request packet fields can be provided via the DMY. **Conversely, when a DMY is not used, default CUD and FAC values are used (see below)**.

Note: All supported operands for a TYPE=DMY REMOTE can be specified in the CDF. They can also be changed and/or added dynamically using the MRMT command.

If you set the IPADDR=, CLDADDR=, CLGADDR=, IFN= and VCN= modifiers, they become the default values for the PING command when RNM=*dmyname* is not provided. They will be used if PING is entered with no parameters. Default values can be overridden using positional parameters that follow the remote protocol (XOT is assumed if omitted) where the IP address is the first parameter, followed by the ***called*** DTE address, MCH interface number and MCH virtual circuit number (the last two (2) values are only used when the protocol is XTP).

If PING is entered with no parameters and both IPADDR= and CLDADDR= are null (not set), the command will be rejected. XOT is the default remote protocol unless you specifically enter XTP. You can always override the default IP address and/or ***called*** DTE address by specifically entering them as positional PING command parameters.

Example: If you have set default values and only want to change the ***called*** DTE address, enter **PING *ddd...ddd***.

For XTP only, two (2) additional parameters are required: ***ifn*** selects the actual MCH interface on the IBM router and ***vcn*** selects the logical channel number on the MCH link for the call.

The PING command sends a Call Request packet to the target IP address carrying the ***called*** DTE address that you specify. A ***calling*** DTE address is supplied from the CLGADDR= modifier if one is given, otherwise, the ***calling*** DTE address is null. The facilities and call user data are fixed as follows:

```
facilities: 080100420707430202
cud:        01000000HNASPING
```

The PING (XOT) Call Request packet is not routed through the LOCAL RTEOUT=values for TYPE=XOT REMOTE selection because the PING operation already know the destination remote router IP address.

The target IP address selects the router. The target DTE address selects the MCH serial interface on the router.

Note: The router is typically configured to map the target DTE address to a specific serial interface or to the router clear operation (via '**x25 route *dte-addr* clear**' configuration).

General notes for PING:

1) Right side positional parameters can be used to override default values set by left side command modifiers.

2) XOT is assumed if a protocol is not specified.

3) If an overriding IP address **and** DTE address are both required, the IP address must be specified first.

4) If an overriding IP address **or** DTE address is required, either can be specified by themselves (no comma place holder is needed).

5) The IFN= and VCN= modifiers are only required for XTP routers. If overriding IFN and VCN parameters are required without overriding IP and/or DTE addresses, comma(s) must be used as place holders.

6) If the RNM= modifier identifies a TYPE=DMY REMOTE or if a *dmyname* is provided as the first PING argument, **all** PING operands will come from the specified REMOTE. However, if standard PING arguments are also given, after the *dmyname*, they will override the operands on the given TYPE=DMY REMOTE.

7) When a DMY is specified for the PING command, the DTEADDR, DCEADDR, FAC and CUD operands are used to supply, respectively, the *called* DTE address, the *calling* DTE address, facilities and call user data. If **NONE** is specified for any of these values, the associated field is null in the Call Request packet. The IPADDR= and PORT= operands on the DMY are used to create the TCPIP socket for the Call Request.

You can override some operand values set for DMY by specifying them as PING command arguments. For example, to override the IPADDR and DTEADDR values for a DMY, you could enter, for example:

```
PING dmyname 192.168.1.100(1998) 20360009
```

When the PING is entered **without a DMY**, the IPADDR=, PORT=, DTEADDR= and DCEADDR= values come from, respectively, the associated command modifier: IPADDR= (aaa.bbb.ccc.ddd(port)), CLDADDR= and CLGADDR=. Since there is no modifier for facilities and call user data, these fields default to 080100420707430202 and 01000000HNASPING, respectively.

Whether a DMY is provided or Call Request information comes from modifiers and PING

command arguments, the IFN and VCN come from the IFN= and VCN= modifiers or from positional arguments specified for the PING command.

8) The PING can result in a Call Accept from the network if the call reaches its destination. Most likely, the PING will result in a Clear Request. The Clear Request cause and diagnostic codes can then be used to determine how far the actual calling function has gone. Please refer to the Cisco documentation and HNAS documentation for a list of clear reason and diagnostic codes.

The following responses can be returned by the PING command.

```
NAS2613M PINGING SOCKET=aaa.bbb.ccc.ddd(port) FOR consname (seqno)
NAS2613M         DTEADDR=ddd...ddd DCEADDR=ddd...ddd
NAS2613M         FAC=xx...xxx
NAS2613M         CUD=xx...xxx
```

This message is generated when the PING operation is started if valid values are supplied for the IP address, Port number and *called* DTE address, *calling* DTE address, facilities data and call user data. These values can come form a TYPE=DMY REMOTE that is specified for the PING command and/or override values that are specified with the PING command.

```
NAS2611M PING CALL TIMED OUT FOR SOCKET=aaa.bbb.ccc.ddd(port)
         DTEADDR=ddd...ddd (seqno)
```

This message is generated when no response is received for the PING Call Request packet within a timeout interval fixed at 15 seconds. A timeout normally indicates that the specified IP address is invalid or the target router is not connected to the network.

```
NAS2615M PING CALL CLEARED (cc/dd) FOR SOCKET=aaa.bbb.ccc.ddd(port)
         DTEADDR=ddd...ddd (seqno)
```

This message is generated when a Clear Request packet is received as the response to the PING Call Request packet. The ***cc/dd*** value provides additional information. Please refer to the Cisco documentation and HNAS documentation for a breakdown of the clear cause and diagnostic codes.

```
NAS2617M PING CALL ACCEPTED FOR SOCKET=aaa.bbb.ccc.ddd(port)
         DTEADDR=ddd...ddd (seqno)
```

This message is generated when a Call Accept packet is received as the response to the PING Call Request packet.

Note: PING NAS261*n*M message support was introduced into 240 with APAR 2400022.

Note: PING NAS261*n*M message support was updated to include the correlation sequence number (***seqno***) and IP and DTE addresses in the response messages effective with APAR 2400039.

PRNT Command - SYSPRINT Logging Control (Privileged)

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	PRNT	[{ON|OFF}]		
		[CLOSE]	N	220
		[OPEN *ddname*]	N	220
		[CLSOPN *ddname*]	N	220
		[RSMEPRSW]	N	240
		[{NEXTPRSW|SWITCHNOW}]	N	240
		[DDNAME=?]	N	220
		[DATE {ON|OFF}]	N	230
		[TIMESTAMP={HH:MM:SS[.T[H[M[I]]]]| STD|EXT}[{+D|-D}]	N	240
		[CNFG {ON|OFF}]	N	220
		[LU {ON|OFF}]	N	230
		[QLLC {ON|OFF}]	A	230
		[SYS {ON|OFF}]	N	230
		[TCP {ON|OFF}]	N	230
		[VC {ON|OFF}]	N	230
		[VTAM {ON|OFF}]	N	230
		[XOT {ON|OFF}]	N	230
		[XTP {ON|OFF}]	N	230
		[PRTSWLST={?| *| ({LOOP|STOP} {SWITCHAT*hh*00| SWITCHATMIDNIGHT| SWITCHAT6AM| SWITCHATMIDDAY| SWITCHAT6PM}, {*ddname*1|DYNAMIC=*ocls*}, : {*dd*namen|DYNAMIC=*ocls*})}]	N	240

This command is used to start (**ON**) or stop (**OFF**) SYSPRINT console logging, control SYSPRINT information alert logging by component type (**LU**, **QLLC**, etc.), control SYSPRINT record date stamping (**DATE ON|DATE OFF**), close and open different SYSPRINT log files (**CLOSE|OPEN|CLSOPN**), resume PRTSWLST= processing (**RSMEPRSW**), force a PRTSWLST= DDNAME switch (**NEXTPRSW|SWITCHNOW**), display (**PRTSWLST=?**), clear (**PRTSWLST=***) or update (**PRTSWLST=(*list*)**) the PRTSWLST= array and display the name of the currently active SYSPRINT log file (**DDNAME=?**). Local console output is always written to the **SYSCONS** unless SHOWOFF is in effect.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

PRNT Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
CLSOPN *ddname*	CLOSE|OPEN *ddname*	3
CLOSE	CLSOPN *ddname*	3
OPEN *ddname*	CLSOPN *ddname*	4
DATE ON	DATE OFF	5
DATE OFF	DATE ON	6
QLLC ON	QLLC OFF	7
QLLC OFF	QLLC ON	8
CNFG ON	CNFG OFF	9
CNFG OFF	CNFG ON	10
LU ON	LU OFF	11
LU OFF	LU ON	12
SYS ON	SYS OFF	13
SYS OFF	SYS ON	14
TCP ON	TCP OFF	15
TCP OFF	TCP ON	16
VC ON	VC OFF	17
VC OFF	VC ON	18
VTAM ON	VTAM OFF	19
VTAM OFF	VTAM ON	20
XOT ON	XOT OFF	21
XOT OFF	XOT ON	22
XTP ON	XTP OFF	23
XTP OFF	XTP ON	24
TIMESTAMP=*format*[{+D|-D}]	none	25
RSMEPRSW	none	26
NEXTPRSW|SWITCHNOW	none	27
PRTSWLST=*|SWITCH=*	none	28
PRTSWLST=*list*|SWITCH=*list*	none	29
DDNAME=?	none	30

PARAMETER	CONFLICT	ORDER
PRTSWLST=?|SWITCH=?	**none**	**31**

- Enter **TIMESTAMP=HH:MM:SS[.T[H[M[I]]]]** to set the HNAS SYSPRINT log timestamp format thus forcing the resolution is to be seconds, tenths of seconds (T), hundredths of seconds (H), thousandths of seconds (M) or ten-thousandths of seconds (I). **STD** and **EXT** are abbreviations for **HH:MM:SS** and **HH:MM:SS.THMI**, respectively.

  Append **+D** to the TIMESTAMP format to cause the Julian Date to prefix the timestamp for each SYSPRINT record (treated the same as PRNT DATE ON).

  Append **-D** to the TIMESTAMP format to prevent the Julian Date from prefixing the timestamp for each SYSPRINT record (treated the same as PRNT DATE OFF).

- Enter **PRNT PRTSWLST=({LOOP|STOP},**
  {SWITCHAT*hh*00|
  SWITCHATMIDNIGHT|SWITCHAT6AM|
  SWITCHATMIDDAY|SWITCHAT6PM},
  {*ddname*m|DYNAMIC=*ocls*},...,{*ddname*n|DYNAMIC=*ocls*})

  to update PRTSWLST= operand actions and/or list entries of static (*ddname*i) or dynamic (DYNAMIC=*ocls*) DDNAMEs that permit automatic SYSPRINT switching when the current SYSPRINT log file becomes full or when the designated action occurs. You may specify static DDNAMEs and/or request DYNAMIC DDNAME allocation in any combination. The DDNAMEs you specify or request dynamically are used sequentially. The default SYSPRINT file is always used initially (DDNAME=SYSPRINT).

  When static DDNAMEs are specified in the PRTSWLST= operand list, they must be defined in the HNASXEQ JOB otherwise the following error message will be issued:

  ```
  NASC211E ddnamei LOG FILE COULD NOT BE OPENED, REQUIRED
  ```

  If you specify DYNAMIC= (no output class) or DYNAMIC (no equal sign), class A is assumed.

  You can also specify just one DYNAMIC value with the LOOP action which will cause a new dynamically allocated SYSOUT file to be used when the PRTLMT is reached, the specified switch action event occurs or when the PRNT NEXTPRSW console command is entered. For example, `PRTSWLST=* PRTSWLST=(LOOP,SWITCHATMIDNIGHT,DYNAMIC).`

  The dynamic datasets are allocated with a DDNAME of DYN#*xxxx* where *xxxx*=0001 for the first dynamically allocated dataset, *xxxx*=0002 for the second and so on. *xxxx* is incremented by one as new datasets are allocated. If you issue a DNAS JCL command, you will see the dynamically allocated DDNAMEs listed. This DDNAME naming convention was chosen so the dynamic DDNAMEs would be easy to remember.

  Note: If you specify a DDNAME of the form DYN#*xxxx* (where *xxxx*=0001, 0002, etc.) in the HNASXEQ JOB, the following alarm message will be generated:

```
NAS0210E DDNAME(S) OF THE FORM DYN#xxxx WERE SPECIFIED, PRTSWLST=DYNAMIC
NAS0210E WILL NOT BE ALLOWED, DDNAME(S) CONFLICT
```

This message is also issued if a PRTSWLST= switch is requested for a DYNAMIC entry. The reason for this message is that DDNAME(s) of this form are reserved for PRTSWLST=DYNAMIC support and thus cannot be specified as static DDNAMEs in the HNAS start JCL. Note also that the DYNAMIC entry is skipped and the next non-DYNAMIC real DDNAME in the PRTSWLST= operand list is used. If there are no real DDNAMEs in the PRTSWLST= operand list, SYSPRINT logging is terminated (the STOP action is forced).

Note: PRTSWLST=DYNAMIC support was introduced into 240 as Enhancement APAR 2400001.

LOOP, STOP, SWITCHAT*hh*00, SWITCHATMIDNIGHT, SWITCHAT6AM, SWITCHATMIDDAY and SWITCHAT6PM are reserved keywords that can be specified anywhere in the PRTSWLST= operand list. They are NOT treated as DDNAMEs.

- **LOOP** specifies that the DDNAMEs in the PRTSWLST= operand list are to be used in a round robin fashion. When the default SYSPRINT file becomes full, it is closed and *ddname*1 is then used. When *ddname*1 becomes full, *ddname*2 is then used. This continues until *ddname*n (the last DDNAME in the list) is used, in which case the *ddname*1 file is reused and the process repeats. Note that the default SYSPRINT file is not reused unless SYSPRINT is specified as a *ddname*i entry in the PRTSWLST= operand list.

- **STOP** specifies that SYSPRINT logging stops when the last DDNAME (*ddname*n) in the PRTSWLST= operand list becomes full.

- **SWITCHAT*hh*00** (military time) specifies that logging in the current PRTSWLST= dataset is to stop and logging in the *ddname*i dataset is to start at the hour specified by *hh* every day. Note that *hh* can be 01 to 24|00 (00 is treated the same as 24).

- **SWITCHAT6AM** (=SWITCHAT0600) specifies that logging in the current PRTSWLST= dataset is to stop and logging in the next PRTSWLST= *ddname*i dataset is to start at 6AM every day.

- **SWITCHATMIDDAY** (=SWITCHAT1200) specifies that logging in the current PRTSWLST= dataset is to stop and logging in the next PRTSWLST= *ddname*i dataset is to start at 12PM every day.

- **SWITCHAT6PM** (=SWITCHAT1800) specifies that logging in the current PRTSWLST= dataset is to stop and logging in the next PRTSWLST= *ddname*i dataset is to start at 6PM every day.

- **SWITCHATMIDNIGHT** (=SWITCHAT2400) specifies that logging in the current PRTSWLST= dataset is to stop and logging in the next PRTSWLST= *ddname*i dataset is to start at 12AM every day.

  Note: If the STOP action is in effect, logging is terminated when a switch occurs. Switching can occur based on a PRTSWLIST= action like SWITCHATMIGNIGHT or when the current

PRTSWLST= *ddname*i dataset becomes full. For this reason, we recommend that the LOOP action be used when SWITCHAFTERINIT or any of the SWITCHAT*time* actions is specified.

Example: `PRNT PRTSWLST=(LOOP,SWITCHATMIGNIGHT,SYSPRNT1,SYSPRNT2,SYSPRNT3).`

You may enter up to 16 DDNAMEs.

The PRNT console command processor will append PRTSWLST= DDNAME values to the existing PRTSWLST= array rather than resetting the array first. This eliminates the need to enter an entire list of DDNAMEs in order to add just one entry to the end of the list. For example, if PRTSWLST=(LOOP,SYSPRNT1) was specified on the BUILD definition statement in the CDF and if PRNT PRTSWLST=(SYSPRNT2) is then issued as a console command, The **PRNT PRTSWLST=?** command would produce the following display:

```
PRNTDDNM=SYSPRINT   (OPEN)
PRTSWLST=LOOP       (INACTIVE)
          SWITCHATMIGNIGHT
          SYSPRNT1
          SYSPRNT2
```

The default SYSPRINT DDNAME should not be included in the list unless you want it to be reused. If a list entry is reused after it was detected full and DISP=MOD is specified, it will appear full again immediately when the first record is written because new data is added to the end of the file. DISP=OLD will cause the old data to be purged so that new data is logged at the beginning of the file when it is reused.

In all cases, an alert message is issued when a SYSPRINT switch takes place and, in the case of the STOP action, when the last PRTSWLST= DDNAME (*ddname*n) file is used. This will indicate that SYSPRINT logging has been terminated. If you wish to restart SYSPRINT logging, you will manually have to enter the PRNT OPEN *ddname* command or the new PRNT RSMELIST command.

- Enter **PRNT PRTSWLST=*** to force the PRTSWLST= array to be cleared. The array count is set to zero, all actions and DDNAMEs are set to null values.

  If you wish to clear the PRTSWLST= array before entering new values, enter PRNT PRTSWLST=* to clear the entries followed by PRTSWLST=({LOOP|STOP},...,*ddnamelist*) to set new actions and DDNAME entries.

- Enter **PRNT PRTSWLST=?** to display the PRTSWLST= actions and DDNAMEs that are currently in effect. The following is example of the information produced by the PRNT PRTSWLST=? command when

  `PRTSWLST=(LOOP,SWITCHATMIGNIGHT,SYSPRNT1,SYSPRNT2,SYSPRNT3)`

  has been specified on the BUILD definition statement or when

  `PRNT PRTSWLST=* PRTSWLST=(LOOP,SWITCHATMIGNIGHT,SYSPRNT1,SYSPRNT2,SYSPRNT3)`

  has been entered as a console command:

```
PRNTDDNM=SYSPRINT   (OPEN)
PRTSWLST=LOOP       (INACTIVE)
         SWITCHATMIDNIGHT
         SYSPRNT1
         SYSPRNT2
         SYSPRNT3
```

- To summarize, you may enter a PRNT parameter that requests a specific action to be performed as follows:

PRNT Action Matrix

PARAMETER	ACTION
ON or omitted	**Write output to the SYSPRINT log file.**
OFF	**Do not write output to the SYSPRINT log file.**
CLOSE	**Close the current SYSPRINT log file. If it the SYSPRINT DD statement specifies DISP=SHR|OLD, all data will be lost if the same SYSPRINT log file is reopened. If the SYSPRINT DD statement specifies DISP=MOD, future logging will resume where it left off when the dataset was closed. Note that if the SYSPRINT dataset is the JES SYSOUT=* spool, DISP=MOD is the default so no data is lost if the dataset is closed and reopened.**
OPEN *ddname*	**Open a new SYSPRINT log file. If a SYSPRINT log file is already opened, an error message is generated. The current SYSPRINT log file must be closed before a new one can be opened.**
CLSOPN *ddname*	**Open a new SYSPRINT log file after the old one is closed. PRNT CLSOPN *ddname* provides the same function as PRNT CLOSE followed by PRNT OPEN *ddname*.**
RSMEPRSW	**Resume BUILD PRTSWLST= operand processing that has been STOPped.**
NEXTPRSW|SWITCHNOW	**Force switch to next DDNAME in BUILD PRTSWLST= operand list.**
PRTSWLST={?|*|(*list*) or SWITCH={?|*|(*list*)}	**Display, clear or update BUILD PRTSWLST= operand array.**
DDNAME=?	**Display current SYSPRINT DDNAME.**
DATE [ON]	**Prefix each record written to the SYSPRINT log file with a Julian Date. Date format provided is *yy.dte* (i.e. 06.001 for 2006, January 1st)**
DATE OFF	**Do not prefix each record written to the SYSPRINT log file with the Julian Date.**
TIMESTAMP=*format*[{+D|-D}]	**Set the SYSPRINT log timestamp format and, optionally, activate or deactivate the Julian Date prefix.**
CNFG [ON]	**Allow configuration information alert messages (NAS1*xxx*I) to be written to the SYSPRINT log file.**

PARAMETER	ACTION
CNFG OFF	Do not allow configuration information alert messages to be written to the SYSPRINT log file.
LU [ON]	Allow LU information alert messages (NAS4*xxx*I) to be written to the SYSPRINT log file.
LU OFF	Do not allow LU information alert messages to be written to the SYSPRINT log file.
QLLC [ON]	Allow QLLC information alert messages (NAS8*xxx*I) to be written to the SYSPRINT log file.
QLLC OFF	Do not allow QLLC information alert messages to be written to the SYSPRINT log file.
SYS [ON]	Allow system information alert messages (NAS0*xxx*I and NAS9*xxx*I) to be written to the SYSPRINT log file.
SYS OFF	Do not allow system information alert messages to be written to the SYSPRINT log file.
TCP [ON]	Allow TCP/IP information alert messages (NAS2*xxx*I) to be written to the SYSPRINT log file.
TCP OFF	Do not allow TCP/IP information alert messages to be written to the SYSPRINT log file.
VC [ON]	Allow VC information alert messages (NAS5*xxx*I and NASA*xxx*I) to be written to the SYSPRINT log file.
VC OFF	Do not allow VC information alert messages to be written to the SYSPRINT log file.
VTAM [ON]	Allow VTAM information alert messages (NAS3*xxx*I) to be written to the SYSPRINT log file.
VTAM OFF	Do not allow VTAM information alert messages to be written to the SYSPRINT log file.
XOT [ON]	Allow XOT information alert messages (NAS7*xxx*I) to be written to the SYSPRINT log file.
XOT OFF	Do not allow XOT information alert messages to be written to the SYSPRINT log file.
XTP [ON]	Allow XTP information alert messages (NAS6*xxx*I) to be written to the SYSPRINT log file.
XTP OFF	Do not allow XTP information alert messages to be written to the SYSPRINT log file.

General Notes for PRNT:

1) HNAS activates with **PRNTON**, **PRNTCNFG**, **PRNTLU**, **PRNTQLLC**, **PRNTSYS**, **PRNTTCP**, **PRNTVC**, **PRNTVTAM**, **PRNTXOT** and **PRNTXTP** set in the absence of any **PRNTxxx** start parameter.

2) The **PRNTON**, **PRNTOFF**, **PRNTCNFG**, **PRNTLU**, **PRNTQLLC**, **PRNTSYS**, **PRNTTCP**, **PRNTVC**, **PRNTVTAM**, **PRNTXOT** and **PRNTXTP** start parameters perform the same functions as the **PRNT [ON]**, **PRNT OFF**, **PRNT CNFG**, **PRNT LU**, **PRNT QLLC**, **PRNT SYS**, **PRNT TCP**, **PRNT VC**, **PRNT VTAM**, **PRNT XOT** and **PRNT XTP** console commands, respectively, for *global* SYSPRINT control. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

3) If multiple SYSPRINT log files are to be opened and closed, we recommend using DISP=SHR so that the closed files can be manipulated while HNAS is running.

When the HNASXEQ job specifies multiple SYSPRINT datasets, each can be used serially (but not in parallel) via the PRNT CLSOPN *ddname* console command. If you plan to 'look in' a dataset (a DASD file and not just the JES SYSOUT=* spool) while HNAS is running, you must specify DISP=SHR for these datasets in the HNASXEQ job. This will allow you to examine and manipulate the dataset while HNAS is running after it is closed via the PRNT CLSOPN *newddname* console command. If DISP=OLD|MOD is specified, these datasets will not be available until HNAS is terminated. When DISP=OLD|SHR is specified for a DASD dataset, old data will be deleted if the dataset is closed and reopened. If DISP=MOD is specified, data will be added to the end of the dataset if it is closed and reopened but as mentioned, it cannot be examined until HNAS is terminated. The exception to this rule is when SYSPRINT is the JES SYSOUT=* spool. In this case, DISP=MOD is the default and SYSPRINT can always be examined and saved via the SDSF DA or ST menu.

4) If the number of records written to the SYSPRINT data set reaches the limit specified by the PRTLMT= operand on the BUILD definition statement, HNAS will stop logging output. The SYSPRINT data set can be the JES spool, an z/OS, OS/390 or MVS sequential data set. A message will be written to the local console each time 4096 records are lost.

5) If the PRTSWLST=({LOOP|STOP},SWITCHAT*time*,*ddname*1,...,*ddname*n) operand is provided on the BUILD definition statement, a secondary log file will automatically be opened when the current log file is closed due to the PRTLMT= value being reached, when the dataset is detected as being full or at the specified time. The PRTSWLST= operand allows multiple log files to be specified so that each can be used successively as the current file is filled. The LOOP|STOP keyword suboperands of the PRTSWLST= operand provide an action of what HNAS should do when the end of the list is detected. LOOP indicates that the PRTSWLST= function should repeat starting with the first PRTSWLST= file (*ddname*1). STOP indicates that logging should be stopped when the last PRTSWLST= file (*ddname*n) is filled. In this case, you would have to manually restart the PRTSWLST= function using the PRNT OPEN *ddname* or PRNT RSMEPRSW command.

6) You may enter **PRTSWLST=** or **SWITCH=** as a PRNT command operand. These keywords are treated the same and their list values are processed identically.

7) When **PRTSWLST=?** is entered, the current SYSPRINT DDNAME is displayed followed by the PRTSWLST= operand actions and DDNAMEs. In addition, the display will show the state of the current SYSPRINT DDNAME and PRTSWLST= array. For example:

```
PRNTDDNM=ddname    (ddnmstate)
PRTSWLST=LOOP      (prswstate)
         SWITCHATMIDNIGHT
         SYSPRNT1 (curddnm)
         SYSPRNT2
         SYSPRNT3
```

Where: *ddname* = current SYSPRINT DDNAME
ddnmstate = OPEN|CLOSED
prswstate = INACTIVE => switch has not occurred - default SYSPRINT is being used
ACTIVE => switch has occurred
STOPPED => STOP action was specified and last DDNAME was used
SUSPENDED => PRNT CLOSE issued when PRTSWLST= list was active
curddnm = marks the DDNAME that is currently in use when *prswstate*=ACTIVE or the DDNAME that was last used when *prswstate*=STOPPED|SUSPENDED.

Note: When PRNTDDNM=*ddname* identifies a DDNAME that is CLOSED, SHOW CONS ON must be active so that the PRTSWLST=? display will be directed to SYSCONS.

8) Forced alarm messages **NAS0207W**, **NAS0208I** and **NAS0209E** are now generated when PRTSWLST= SYSPRINT switching support is enabled. Please refer to the Alert/Alarm Message and Clear/Reset Code Enhancements area of this section for a description of the messages.

9) Forced alarm messages **NAS0210I**, **NAS0211W** or **NAS0230W** are now generated, respectively, when the PRNT ON, PRNT OFF or PRNT DATE ON|OFF TCP ON|OFF ... commands are entered to inform the SYSCONS operator of changes to SYSPRINT logging. Please refer to the Alert/Alarm Message and Clear/Reset Code Enhancements area of this section for a description of the messages.

Q or QUIT Command - Terminate Operations

(changed for V2R2M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	Q[UIT]			
	QA[/*pswd*]		**C**	**220**
	QY[/*pswd*]		**C**	**220**
	QQ[/*pswd*]		**C**	**220**
	QS		**N**	**220**
	QE	***ddname*|(*cmd*1,...,*cmd*n)**	**N**	**240**
	QN			

This command is used by the local console operator to shutdown HNAS and by a remote console operator to terminate his console session.

The **Q** (QUIT) command, entered by itself, generates the following prompt message.

```
NASC000I ENTER: N=RESUME, Y=END, E=EXEC, S=STOP, Q=QUICK, A=ABEND
```

To bypass the prompt, you may enter the response as a follower to the **Q** command (e.g., **QS**).

- **For remote console sessions**, you may reply with **N** or **Y**.

  N causes the remote console session to resume with a prompt for a new command.

  Y causes the remote console session to be terminated and an XTP or XOT Clear Request packet to be transmitted to the remote router.

  All other followers are an invalid reply from a remote console operator.

- **For local console sessions**, you may reply with **N**, **Y, Q**, **A**, **S** or **E**.

  N causes the local console session to resume with a prompt for a new command.

  Y causes HNAS to shutdown resulting in all TCP/IP and VTAM connections being quiesced then severed (graceful termination) with a SNAP dump of all HNAS control blocks to be formatted in SYSPRINT. The **Y** reply is useful after conducting tests where control block review is required.

  Q is treated the same as **Y** except that a SNAP dump of HNAS control blocks is not taken making shutdown faster and less CPU intensive. The **Q** reply is considered the normal shutdown method.

A causes HNAS to ABEND resulting in fast termination of all TCP/IP and VTAM connections (abrupt termination). An HNAS storage dump is also produced.

Note: The **Y**, **Q** and **A** replies from a local console operator will require a password exchange to complete the operation if the CONPSWD= operand was specified on the BUILD definition statement. You may enter the console password with the reply (***lpswd***) to avoid a separate password prompt and password exchange.

The **S** reply is the same as the **Q** reply except no password is required even if one was specified for the CONPSWD= operand.

The **E** reply is used to schedule the execution of a command list prior to shutdown. The command list can be provided in a file which is identified by the ***ddname*** argument or can be provided on the same line as the QE command via the **(*cmd*1,...,*cmd*n)** argument. Note that no prompt is provided for the QE command argument. if QE is entered without a *ddname* or inline command list, it is processed as though QS were entered. The following is a sample command list that can be provided via **QE *ddname***:

```
* THIS IS A SAMPLE CONSOLE COMMAND LIST FILE. COMMENTS CAN
* BE SUPPLIED THAT START WITH '*' OR ';' IN RECORD COLUMN 1.
* COMMENTS ON INDIVIDUAL LINES CAN BE SUPPLIED THAT START
* WITH ';'.  CONSOLE COMMANDS WILL BE EXTRACTED FROM EACH
* RECORD FROM THE FIRST NON-BLANK CHARACTER TO THE LAST
* NON-BLANK CHARACTER ON EACH LINE.  THE SEARCH FOR THE
* FIRST NON-BLANK CHARACTER IS EXECUTED FOR NON-COMMENT
* RECORDS ONLY (*|; IS NOT IN CC1).  THE SEARCH FOR THE
* LAST NON-BLANK CHARACTER IS AFFECTED AFTER THE LINE
* COMMENT STARTING DELIMITER (;) IS FOUND.
* |<-- FIRST NON-BLANK
* |
* |    |<-------- LAST NON-BLANK
* |    |  |
* V    V  V
  DPARM             ; 1ST QUEUED COMMAND TO DISPLAY FINAL HNAS PARMS
  DNAS JCL          ; 2ND QUEUED COMMAND TO DISPLAY FINAL HNAS JCL
```

The SYSPRINT log now contains the HNAS global parameters in effect at shutdown as well as the final HNAS DDNAMEs that were allocated either statically or dynamically (see PRTSWLST=DYNAMIC support on page CONS-213 for details). For more information on command list processing, see Input Rules, Command Lists on page CONS-9 and the EXEC command description or page CONS-139.

Note: The **ALARM LOG=?** command is executed automatically when HNAS is SHUTDOWN using the Q (QUIT) command. This is done unconditionally for any QUIT command follower. If QE *ddname* is entered, the ALARM LOG=? command is executed before the command list identified by *ddname*.

Note: The **N**, **Y, Q**, **A**, **S** and **E** followers are abbreviations for **NO**, **YES**, **QUICK**, **ABEND**, **STOP** and **EXEC** which can also be entered. For example, **QS** and **QSTOP** are treated identically.

- **Shutdown status messages**:

  When HNAS terminated with **QQ**, **QS** or **QE**, the following message is displayed.

  NAS0030I SHUTDOWN STARTED, RELEASING RESOURCES

  When HNAS terminated with **QY**, the following message is displayed.

  NAS0031I SHUTDOWN STARTED, FORMATTING STORAGE, RELEASING RESOURCES

  When HNAS terminated with **QA**, the following message is displayed.

  NAS0032I SHUTDOWN STARTED, FORCED ABEND REQUESTED

  For all HNAS termination requests except **QA**, the following message is displayed when shut-down is complete.

  NAS0035I SHUTDOWN COMPLETED, RC=*retcode*

RMTCONS Command - Remote Console Control (Privileged)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	RMTCONS	**[{ON|OFF|PRIV}]**		

This command is used to enable (**ON|PRIV**) or disable (**OFF**) HNAS remote console access. The command is provided so that data center operations personnel can control remote access to the HNAS Console Subsystem for security reasons.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

RMTCONS Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**
PRIV	**none**	**3**

- Enter **RMTCONS [ON]** to enable remote console access. This will be required for remote HNAS product support. Remote console access **can** and **should** be password protected. For more information, see description of the CONPSWD operand of the BUILD definition statement in Chapter 4 of the HNAS Guide and Reference documentation.

- Enter **RMTCONS PRIV** to enable privileged remote console access. The remote console operator can **select privileged status by entering the console password *backwards***. Privileged remote console access **must** be password protected. A console password that is the same when entered backward or forwards (e.g., CONPSWD=WOW), will prevent privileged status from being attained. In this case, remote console access will be 'display only'. Privileged remote consoles can enter commands that affect HNAS operation (e.g., TRCDISP ON) while non-privileged remote consoles cannot. HNAS diagnostic support may require privileged remote console access. This command allows data center personnel to control who may access their system remotely and what they can do.

  Note: When CONLMT=0 is specified on the BUILD definition statement, the RMTCONS ON|PRIV command will be rejected with message '**NASC511E PARAMETERS CONFLICT, REQUIRED**' because no console control blocks were created to support remote console sessions.

- Enter **RMTCONS OFF** to inhibit remote console access. An attempt to connect to the HNAS Console Subsystem from a remote location will result in the transmission of an XTP or XOT Clear Request packet.

General Notes for RMTCONS:

1) HNAS activates with **RMTCONS OFF** set in the absence of the **RMTCONS** or **RMTCONP** start parameters.

2) The **RMTCONS [ON]** and **RMTCONP** start parameters perform the same function as the **RMTCONS [ON]** and **RMTCONS PRIV** console commands, respectively, for REMOTE console enabling. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

RNM= Modifier - Set REMOTE Name

(changed for V2R2M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
RNM=*rmtname* **RNM=?**	***command***	**[RNM=*rmtname*]**	**N** **C**	**114** **240**

This modifier is used to select a specific REMOTE definition statement name for the commands that follow. The **RNM=** value must identify a valid REMOTE definition statement in the HNAS Configuration Data File. Use **RNM=** to target a specific client resource.

Note: The REMOTE resource named by the RNM= modifier is used by the DLU, DMCH, DNWDF, DPCE, DRMT, DSTATS, DVC, MON, MRMT, PING, STATS, TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCLUQ, TRCMCH, TRCMCHX, TRCPCE, TRCVC, TRCVCQ and VARY RMT commands.

- When a ***rmtname*** value is specified, **RNM=** takes precedence over the **ID=** and **IFN=** value combination. If the **RNM=** modifier is entered with no ***rmtname*** value, the remembered REMOTE name is reset.
- Enter **RNM=?** to display the permanent RNM= value.

SCHEDULE Command - Schedule Commands for Execution

(new for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	SCHEDULE	[{**LCLCONS**|*rconname*}]	**A**	**240**
		[{*ddname*| (*time*1,*cmd*1,...,*time*n,*cmd*n}] [**LIST** [*ddname*]] [{**SUSP**|**SUSPEND**}] [{**RSME**|**RESUME**}] [{**PRG**|**PURGE**}]	**A**	**240**

This command is used to execute a schedule list file identified by ***ddname*** or supplied as an *inline* schedule list **(*time*1,*cmd*1,...,*time*n,*cmd*n)**, to display a schedule list file identified by ***ddname*** (**LIST *ddname***), to display the active (RESIDENT) schedule list (**LIST**), to suspend the active schedule list (**SUSP|SUSPEND**), to restart the suspended schedule list (**RSME|RESUME**) or to delete the active schedule list (**PRG|PURGE**).

- When the **LCLCONS** or ***rconname*** argument is provided, it directs SCHEDULE processing to the identified console. This allows one console to alter SCHEDULE processing for another console. For example, SCHEDULE RCON0031 (12:00:00,DNAS,13:00:00,ALARM SHOW) will cause the DNAS command to be executed at noon and the ALARM SHOW command to be executed at 1pm for the remote console named RCON0031 regardless of which console actually entered the command. For more information on LCLCONS|*rconname* processing, please refer to section entitled 'Console PCE name as a Suffix Command Modifier' on page CONS-11.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

SCHEDULE Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
SUSP|SUSPEND	**RSME|RESUME**	**LCLCONS|*rconname***	**1**
RSME|RESUME	**SUSP|SUSPEND**	**LCLCONS|*rconname***	**2**
PRG|PURGE	**none**	**LCLCONS|*rconname***	**3**
ddname	**(*time*1,*cmd*1,...,*time*n,*cmd*n)**	**LCLCONS|*rconname***	**4**
(*time*1,*cmd*1,...,*time*n,*cmd*n)	***ddname***	**LCLCONS|*rconname***	**5**
LIST	**none**	**LCLCONS|*rconname***	**6**

- Enter **SCHEDULE *ddname*** to execute a schedule list file identified by *ddname*. *ddname* identifies a DDNAME JCL statement that exists in the HNAS start job. DDNAME, in turn, points a file that contains a list of times and console commands. The DDNAMEs you that provide in the HNAS start job are arbitrary: For example:

```
//SCHDLST1 DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST1),DISP=SHR
//SCHDLST2 DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST2),DISP=SHR
//SCHDLST3 DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST3),DISP=SHR
//SCHDLST4 DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST4),DISP=SHR
:
```

These schedule list files can be executed using the SCHEDULE console command as follows:

```
SCHEDULE SCHDLST1
SCHEDULE SCHDLST2
SCHEDULE SCHDLST3
SCHEDULE SCHDLST4
```

If a *ddname* is specified that does not exist in the HNAS start JCL, the following message is issued:

```
NASC051E SCHEDULE ddname INVALID, CANNOT BE OPENED
```

If a *ddname* identifies a DDNAME that references a member of a partitioned dataset and the member does not exist, the following message is issued:

```
NASC204E ddname ABEND PARMLIST=01301886 0008A8B4 008CC410 00000000
```

When a valid schedule list *ddname* is specified and it has been read successfully, the following message is generated:

```
NASC055I SCHEDULE ddname FILE HAS BEEN READ, IT WILL NOW BE SCHEDULED
```

Note: Each time **SCHEDULE *ddname*** is executed, the identified schedule list data will be appended to the end of the active schedule list.

A schedule list file can contain another SCHEDULE *ddname* statement. When the embedded SCHEDULE command is decoded, the balance of the current schedule list will be ignored. The new schedule list will replace the active schedule list at the time associated with the embedded SCHEDULE command. This allows you to chain schedule lists.

You can also chain schedule lists via JCL by concatenating the schedule list dataset names as long as the concatenation forms a sequential dataset. For example:

```
//SCHDLSTS DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST1),DISP=SHR
//         DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST2),DISP=SHR
//         DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST3),DISP=SHR
//         DD DSN=COMM1.V2R4M0.TEMPMAC(SCHDLST4),DISP=SHR
```

These schedule list files can be executed using the SCHEDULE console command as follows:

```
SCHEDULE SCHDLSTS
```

Note: In the initial development, the schedule lists are always sequential datasets or members of partitioned datasets. The SCHEDULE command does not currently allow you to specify a DDNAME and MEMBER name if DDNAME references a PDS.

Note: All DDNAMEs and the members they reference are completely arbitrary. The only requirement is that the *ddname* you provide for the SCHEDULE command ***must*** exist in the HNAS start JCL.

Note: If DISP=SHR is specified for all schedule list DDNAMEs above, it will allow you to modify the schedule lists while HNAS is running.

The commands you provide in a schedule list file are completely arbitrary. The SCHEDULE command processor does not check the commands for validity. Validity checking is provided by the individual command processors.

A schedule list file can contain comment records that start with an asterisk (*) or semi-colon (;) in record column 1. For data records in a schedule list file, a comment can be provided that follows the command text if the first character of the comment is a semi-colon (just like the CDF parsing).

A single time and command can be specified on a single data record as follows:

(*hh*:*mm*:*ss* , *cmd*) ; *comment*

Multiple times and commands can also be specified on a single record as follows:

(*hh*:*mm*:*ss* , *cmd* , ... , *hh*:*mm*:*ss* , *cmd*) ; *comment*

Command text is assumed to start with the first non-blank character after the comma that separates the schedule time and command of a data record and end with the last non-blank character of the record (prior to a comma separator or closing parenthesis). Times are given in *hh*:*mm*:*ss* notation where *hh*=00-23, *mm*=00-59 and *ss*=00-59 (00:00:00 is midnight and 12:00:00 is noon).

If an END statement or embedded SCHEDULE command is decoded, saving of commands is terminated. Any commands the follow an END statement or SCHEDULE command are ignored.

Scheduled Command Savearea Size

The maximum number of times and commands that can be saved depends on the available schedule storage space. A total of 512 bytes are reserved for saved times and console commands. Each blank is counted as a text character. The 512 byte maximum also includes 6-bytes for the time and a length byte for each saved command. For example, if SCHEDULE *ddname* referenced a file containing the following:

```
06:00:00 ,  DPARM EXEC        ; execute first command at 6am
```

```
12:00:00 ,  ALARM LOG=?      ; execute second command at noon
01:00:00 , TRCPCE ALLON      ; execute third command at 1am
02:00:00 , TRCPCE ALLOFF     ; execute fourth command at 2am
```

74 bytes would be required to save the commands: 17 for DPARM EXEC (6+1+10), 18 for ALARM LOG=? (6+1+11), 19 for TRCPCE ALLON (6+1+12) and 20 for TRCPCE ALLOFF (6+1+13). If you provide data in a schedule list that exceeds the 512 byte limit, the following message will be issued:

```
NASC054E SCHEDULE XXXXXXXX COMMAND QUEUE LIMIT count/limit,
                  EXCEEDED, SCHEDULE COMMAND ABORTED
```

Where: *count* = required number of bytes
limit = 512 (currently)

Note: 6 bytes are required to save the time value (*hh*:*mm*:*ss*) because an upper window edge (UWE) time is also saved. The UWE time is computed by adding 5 seconds to the specified time. The UWE time is required to provide a time window during which the scheduled command can be executed. Exact times can never actually be used because scheduled commands are processed by the HNAS interval timer subtask which is imprecise because of system task dispatching and task switching.

Note: The CONCMDQ queue area in the console PCE is not used to save scheduled commands but is used when a saved scheduled command is actually scheduled. A separate queue area is used to remember scheduled commands.

Note: The BUILD CONCMDQ= operand can include the **SCHEDULE *ddname*** command.

Note: The DCB parameters for datasets identified by the **SCHEDULE *ddname*** command should be RECFM=FB and LRECL=80 with BLKSIZE= as a multiple of LRECL=.

Listed below is a sample schedule list file:

```
* THIS IS A SAMPLE CONSOLE SCHEDULE LIST FILE. COMMENTS CAN
* BE SUPPLIED THAT START WITH '*' OR ';' IN RECORD COLUMN 1.
* COMMENTS ON INDIVIDUAL LINES CAN BE SUPPLIED THAT START
* WITH ';'.  CONSOLE COMMANDS WILL BE EXTRACTED FROM EACH
* RECORD FROM THE FIRST NON-BLANK CHARACTER TO THE LAST
* NON-BLANK CHARACTER THAT FOLLOWS THE TIME/COMMAND COMMA
* SEPARATOR ON EACH LINE.  THE SEARCH FOR THE FIRST NON-BLANK
* CHARACTER IS EXECUTED FOR NON-COMMENT RECORDS ONLY (*|; IS
* NOT IN CC1).  THE SEARCH FOR THE LAST NON-BLANK CHARACTER IS
* AFFECTED AFTER THE LINE COMMENT STARTING DELIMITER (;) IS
* FOUND.
*             |<-- FIRST NON-BLANK
*             |
*             |         |<-------- LAST NON-BLANK
*             |         ||||
*            V         VVVV
```

```
06:00:00 , DPARM EXEC       ; FIRST SCHEDULED COMMAND
12:00:00 , ALARM LOG=?      ; SECOND SCHEDULED COMMAND
01:00:00 , TRCPCE ALLON     ; THIRD SCHEDULED COMMAND
02:00:00 , TRCPCE ALLOFF    ; FOURTH SCHEDULED COMMAND
```

- Enter **SCHEDULE** (***time*1,*cmd*1,...,*time*n,*cmd*n**) to create an *inline* schedule list on a single line without having to invoke a ***ddname***. Inline times and commands are added to the end of the active schedule list if one exists.

  Note: Inline schedule lists are subject to the same save criteria as *ddname* schedule lists. See the section above entitled '**Scheduled Command Savearea Size**' for details.

- Enter **SCHEDULE LIST *ddname*** to display the individual times and commands in the schedule list file identified by *ddname*. If SCHDLST1 identifies a file that contains the sample schedule list described above, the following would be displayed when SCHEDULE LIST SCHDLST1 was entered:

  ```
  NASC055I SCHEDULE SCHDLST1 FILE HAS BEEN READ, IT WILL NOW BE LISTED

  => 06:00:00 DPARM EXEC
  => 12:00:00 ALARM LOG=?
  => 01:00:00 TRCPCE ALLON
  => 02:00:00 TRCPCE ALLOFF
  ```

  Note: SCHEDULE LIST *ddname* displays times and commands only. Comments are not displayed.

- Enter **SCHEDULE LIST** to display the individual times and commands in the active (RESIDENT) schedule list.

- Enter **SCHEDULE SUSP** to suspend the active schedule list. This does not remove the schedule list but prevents any command entries from being scheduled.

- Enter **SCHEDULE RSME** to resume the suspended schedule list. This allows schedule list command entries to be scheduled.

- Enter **SCHEDULE PRG** to delete the active schedule list. This removes all times and commands thus preventing any commands from being scheduled.

  Note: A scheduled command will terminate a running command and the following message will be issued:

  ```
  NASC005W cmdname ABORTED DUE TO OPERATOR INPUT
  ```

  If the aborted command is part of a running command list, the following message will also be displayed:

  ```
  NASC006W ddname COMMAND LIST INTERRUPTED
  ```

The interrupted command list will the resume with the next command after the scheduled command executes.

General notes for SCHEDULE:

1) You may enclose a *cmd* within single or double quotes (e.g., *cmd*, '*cmd*' or "*cmd*" are valid). If single quotes are part of a *cmd*, it can be enclosed in double quotes (e.g., SMSG '*text*' or "SMSG '*text*'" are valid). The enveloping single or double quotes are removed from each *cmd* before the values are saved.

2) Leading and trailing blanks are removed from each *cmd* and *time* before the values are saved.

3) When SCHEDULE *ddname* is specified, an END command will signal the end of data in the referenced file. In addition, an embedded SCHEDULE command will also signal the end of data in the file. Any records that follow these statements will be ignored. This will reduce the likelihood of receiving the NASC054E error message due to the 512 byte queuing limit being reached.

4) The CONCMDQ queue within the console PCE, which is normally used to execute an EXEC *cmdlist*, is also used to execute scheduled commands. For the EXEC command, *cmd* values are enqueued to CONCMDQ in a first in, first out (FIFO) manner. For the SCHEDULE command, *cmd* values are enqueued to CONCMDQ in a last in, first out (LIFO) manner. LIFO enqueue is used because scheduled commands must be executed at the specified time rather than after any other queued commands.

Note: There is an independent CONCMDQ queue for every console PCE (local and remote). The BUILD CONCMDQ= operand is saved in the CONCMDQ queue within the local console PCE only. When a console user issues an EXEC or SCHEDULE command, the CONCMDQ queue in the associated console PCE is used so that the queued command(s) are executed only for the console issuing the EXEC or SCHEDULE command.

WARNING: If a command is currently running when a scheduled command is enqueued, the current command is aborted in favor of the scheduled command (the same action takes place when a command is entered by a console operator). If a command list is running, the list is interrupted while the scheduled command executes. The command list is then restarted with the next queued command after the scheduled command executes.

5) Multiple commands can be scheduled for execution at the same time. For example:

(12:00:00,ALARM LOG=?,12:00:00,DNAS)

WARNING: Because scheduled commands are enqueued to the CONCMDQ queue in a LIFO manner, commands with the same schedule time are executed in the reverse order from how they are listed in the schedule list. In the list above, DNAS will be executed before ALARM LOG=? at approximately noon each day. You should consider this when listing commands in a schedule list.

A simple solution to ensure processing order (avoiding reverse coding) is to sequentially list the commands and step by 1 second as depicted in the following sample:

(12:00:00,ALARM LOG=?,12:00:01,DNAS)

6) Wildcards (*) may be specified for any time digit except the low order seconds digit. When a wildcard character is supplied for a time digit (*hh*:*mm*:*ss*), it will be replaced with the corresponding digit from the current time then the resulting time will be tested against the current time.

Examples:

If (**:**:00,ALARM LOG=?) is a scheduled command, the ** will be replaced by the current hours and minutes values so that ALARM LOG=? will be executed every 60 seconds.

If (**:30:00,ALARM LOG=?) is a scheduled command, the ** will be replaced by the current hours value so that ALARM LOG=? will be executed at half past every hour.

If (1*:00:00,ALARM LOG=?) is a scheduled command, the * will be replaced by the low order hours digit so that ALARM LOG=? will be executed on the hour between 10:00 and 19:00 each day.

If (12:**:00,ALARM LOG=?) is a scheduled command, the ** will be replaced by the current minutes value so that ALARM LOG=? will be executed once per minute between 12:00 and 12:59 each day.

Note: You can specify a wildcard for any time digit except the low order seconds digit (*hh*:*mm*:*s**). The low order seconds digit must be a value between 0 and 9. A wildcard is not allowed because it could cause a scheduled command to be executed every second which would impact HNAS performance.

7) When an *inline* list or *ddname* list is given, entries are added to the end of the active schedule list in effect (the RESIDENT list). If the RESIDENT list contains another SCHEDULE command, the specified schedule list will replace the RESIDENT list. This allows you to chain schedule lists (see Note 8 examples below).

Note: Each console PCE (local and remote) has it's own independent RESIDENT queue as it does a COMCMDQ queue. The RESIDENT queue holds the RESIDENT list while the CONCMDQ queue holds scheduled commands ready for execution.

8) A schedule list that contains an embedded SCHEDULE command allows you to chain schedule lists.

Examples:

If you want to run a different schedule list each day, you could structure them as follows (assume that the SCHEDULE command is entered on Monday):

```
SCHEDULE DD4MON

//DD4MON DD DSN=hlq.HNASMAC(DD4MON),DISP=SHR
//DD4TUE DD DSN=hlq.HNASMAC(DD4TUE),DISP=SHR
//DD4WED DD DSN=hlq.HNASMAC(DD4WED),DISP=SHR
//DD4THU DD DSN=hlq.HNASMAC(DD4THU),DISP=SHR
//DD4FRI DD DSN=hlq.HNASMAC(DD4FRI),DISP=SHR
//DD4SAT DD DSN=hlq.HNASMAC(DD4SAT),DISP=SHR
//DD4SUN DD DSN=hlq.HNASMAC(DD4SUN),DISP=SHR
```

Each schedule list file ends with a SCHEDULE command that points at the next file which will take effect at midnight.

The DD4MON file ends with the following statement: (00:00:00,SCHEDULE DD4TUE)

The DD4TUE file ends with the following statement: (00:00:00,SCHEDULE DD4WED)

:

The DD4SUN file ends with the following statement: (00:00:00,SCHEDULE DD4MON)

In this way, you can operate with a different set of scheduled commands each day and because DISP=SHR is specified for each *ddname*, you can make changes to any of the schedule list files while HNAS is running. Note also that you can alter the normal schedule list flow by simply entering a new SCHEDULE command from the operator console.

As another example, suppose you want to run different schedule lists during the day and night, you could structure them as follows (assume that the SCHEDULE command is entered during the day):

```
SCHEDULE DD4DAY

//DD4DAY DD DSN=hlq.HNASMAC(DD4DAY),DISP=SHR
//DD4NIT DD DSN=hlq.HNASMAC(DD4NIT),DISP=SHR
```

As before, each schedule list file ends with a SCHEDULE command that points at the other file which will take effect at the specified time.

The DD4DAY file ends with the following statement: (00:00:00,SCHEDULE DD4NIT)

The DD4NIT file ends with the following statement: (06:00:00,SCHEDULE DD4DAY)

9) When a scheduled command is executed from the CONSMDQ queue, the normal console command echo prefix is changed from HNASCMD-> to SCHDCMD-> so that scheduled commands can be identified in the SYSPRINT log file.

10) If you enter SCHEDULE *ddname*, the DDNAME must be defined in the EXEC PGM=HNAS start JOB.

Note: The new SCHEDULE command support was introduced into 240 as Enhancement APAR 2400064.

SHOW Command - SYSCONS Display Control (Privileged)

(changed for V2R2M0)
(changed for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	SHOW	**[{ON|OFF|ERR|** **CMSG {ON|OFF}|** **CONS {ON|OFF}|** **ALLON|ALLOFF|** **MORE|LESS}]**	**C** **A** **N** **N** **A**	**220** **240** **230** **230** **230**

This command is used to start (**ON**) or stop (**OFF**) SYSCONS display for **all** asynchronous alarm messages (unsolicited output), to start (**ERR**) SYSCONS display for **error** alarm messages only, to start (**CMSG [ON]**) or stop (**CMSG OFF**) message compression processing, to start (**CONS [ON]**) or stop (**CONS OFF**) SYSCONS display for synchronous console command output (solicited output), to start (**ALLON**) or stop (**ALLOFF**) SYSCONS display for all HNAS output (asynchronous and synchronous) or to request more verbose (**MORE**) or less verbose (**LESS**) alarm/trace information to be displayed on SYSCONS and/or logged in SYSPRINT. HNAS messages are always logged in the SYSPRINT log (if **PRNT ON** is in effect) regardless of the SHOW state.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

SHOW Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**
ERR	**none**	**3**
CMSG ON	**CMSG OFF**	**4**
CMSG OFF	**CMSG ON**	**5**
CONS ON	**CONS OFF**	**6**
CONS OFF	**CONS ON**	**7**
ALLON	**ALLOFF**	**8**
ALLOFF	**ALLON**	**9**
MORE	**LESS**	**10**
LESS	**MORE**	**11**

- Enter **SHOW [ON]** to allow **all** alarm messages to be written to SYSCONS.

- Enter **SHOW OFF** to prevent any alarm message from being written to SYSCONS.

- Enter **SHOW ERR** to allow **only error** alarm messages to be written to SYSCONS. Informational alarm messages are not written to SYSCONS.

  Note: The SHOW ON|OFF|ERR action for alarm messages is restricted to the SYSCONS only. Regardless of the SHOW state, all alarm messages will be logged SYSPRINT if **PRNT ON** is in effect.

  Note: If SHOW OFF and PRNT OFF are both in effect (or PRNTLMT= has been reached), there is no way communicate error or informational messages to the operator. For this reason, we recommend running with SHOW ERR (the default when HNAS is started).

  Note: The SHOW ON|OFF|ERR parameters are mutually exclusive and have no effect on console command output.

  Note: For a discussion of the interaction between **SHOW ERR** (start parameter **SHOWERR**) and the ALRMFLTR= operand, see note under the description ALARM FILTER= command in this document or under the description of the ALRMFLTR= operand in Chapter 4 of the HNAS Guide and Reference.

- Enter **SHOW CMSG [ON]** to **allow all** alarm messages to be compressed (multiple consecutive blanks removed).

- Enter **SHOW CMSG OFF** to **prevent any** alarm message from being compressed.

  Note: The **SHOWCMSG** start parameter or **SHOW CMSG ON** console command allows alarm messages containing variable length text data to be compressed (multiple consecutive blanks are removed).

  Prior to APAR 2400036, alarm messages that contain names or other variable text data can cause multiple blanks to appear in the alarm messages. Removing multiple blanks makes messages look cleaner in SYSPRINT log and on SYSCONS. If the SHOWCMSG parameter is not specified, no compression is performed. All messages appear as they always have. This will allow customers to control the compression process for those who use message filtering tools that depend on fixed message offsets. Consider the following message for example:

  NAS3799I Without Compression (SHOWCMSG OFF):

```
0         1         2         3         4         5         6
01234567890123456789012345678901234567890123456789012345678901234567890123456789
|                                                                               |
NAS3799I LU MCH1PVC  ENDING   SESSION ON MCH MCH1     RMT  CAUSE/DIAG=
000/130 (00/82) DIAGX=0000
|                           |
012345678901234567890123457
8         9
```

NAS3799I With Compression (SHOWCMSG ON):

```
0         1         2         3         4         5         6
01234567890123456789012345678901234567890123456789012345678901234567890123456789
|                                                                               |
NAS3799I LU MCH1PVC ENDING SESSION ON MCH MCH1 RMT CAUSE/DIAG=000/130
(00/82) DIAGX=0000
|                 |
012345678901234567
7         8
```

Note: While message compression does use additional CPU cycles, the affect appears to be minimal. However, HNAS environments with excessive alert message activity should consider measuring the effect upon CPU load with compression enabled (SHOWCMSG ON) and disabled (SHOWCMSG OFF) to see if compression is causing an unacceptable percentage of CPU load. Our in-house testing revealed a CPU seconds difference per WTO for SHOWCMSG ON versus SHOWCMSG OFF to be an additional 0.000011048 seconds per WTO or approximately 0.002846% additional overhead per WTO.

Note: The SHOWCMSG support described above was introduced into 240 with APAR 2400036.

- Enter **SHOW CONS [ON]** to allow all console command output to be written to SYSCONS.
- Enter **SHOW CONS OFF** to prevent console command output from being written to SYSCONS.

  Note: The SHOW CONS ON|OFF action for console command output is restricted to the SYSCONS only. Regardless of the SHOW CONS state, all console output will be logged SYSPRINT if **PRNT ON** is in effect.

  Note: The SHOW CONS ON|OFF parameters have no effect on alarm message output.

- Enter **SHOW ALLON** to allow all alarm messages and console output to be displayed at SYSCONS (equivalent to the SHOWON start parameter or SHOW ON CONS ON console command).
- Enter **SHOW ALLOFF** to inhibit alarm messages and console output from being displayed at SYSCONS (equivalent to the SHOWOFF start parameter or SHOW OFF CONS OFF console command). Regardless of the SHOW state, all alarm messages and console output will be logged SYSPRINT if **PRNT ON** is in effect.
- Enter **SHOW MORE** to allow additional alarm/trace information to be displayed at SYSCONS and/or logged in SYSPRINT (if **PRNT ON** is in effect).
- Enter **SHOW LESS** to inhibit additional alarm/trace information from being displayed at SYSCONS and/or logged in SYSPRINT.

- To summarize, you may enter a SHOW parameter to control which type of HNAS output is delivered to SYSCONS as follows:

SHOW Action Matrix

PARAMETER	ACTION	I	W	E	S	A	C
ERR or omitted	Route error alarm messages to SYSCONS	N	Y	Y	Y	Y	
ON	Route all alarm messages to SYSCONS	Y	Y	Y	Y	Y	
OFF	Withhold all alarm messages from SYSCONS	N	N	N	N	N	
CMSG ON	Allow alarm message compression	Y	Y	Y	Y	Y	
CMSG OFF	Prevent alarm message compression	Y	Y	Y	Y	Y	
CONS ON	Route all console output to SYSCONS						Y
CONS OFF	Withhold all console output from SYSCONS						N
ALLON	Route all alarm messages and console output to SYSCONS	Y	Y	Y	Y	Y	Y
ALLOFF	Withhold all alarm messages and console output from SYSCONS	N	N	N	N	N	N
MORE	Route extended information with alarm messages to SYSCONS	O	OE	OE	OE	OE	
LESS	Withhold extended information with alarm messages to SYSCONS	O	OE	OE	OE	OE	

```
For Table Header: I => informational alarms
                  W => warning alarms
                  E => error alarms
                  S => severe error alarms
                  A => alarms that require user action
                  C => Console command output

For Table Body: N     => NO
                Y     => YES
                O     => When SHOWON is in effect
                E     => When SHOWERR is in effect
                blank => Not Applicable
```

General notes for SHOW:

1) When the SHOWMORE option is effect, alarm/trace information like the following is displayed at SYSCONS and/or logged in SYSPRINT:

```
NAS2711I PCESOCDS=FFFAF10B 03FEFDFF 00000000 00000000
NAS2711I          00000000 00000000 00000000 00000000
NAS2711I          00000000 00000000 00000000 00000000
NAS2711I          00000000 00000000 00000000 00000000
:
```

These messages are subject to alarm filtering (ALRMFLTR=) and limiting (ALRMLMTS=). When the SHOWMORE option is not active (SHOWMORE start parameter omitted or SHOW LESS console command entered), these additional alarm/trace messages are withheld.

2) If the number of records written to the SYSPRINT data set reaches the limit specified by the PRTLMT operand on the BUILD definition statement, HNAS will stop logging output. The SYSPRINT data set can be the JES spool, an Z/OS, OS/390 or MVS sequential data set file. A message will be written to the local console each time 4096 records are lost.

3) HNAS activates with **SHOWERR**, **SHOWCONS OFF**, **SHOWMORE OFF** set in the absence of any **SHOWxxx** start parameters.

4) The **SHOWON**, **SHOWOFF**, **SHOWERR**, **SHOWCONS** and **SHOWMORE** start parameters perform the same function as the **SHOW [ON]**, **SHOW OFF**, **SHOW ERR**, **SHOW CONS** and **SHOW MORE** console commands, respectively, for *global* SYSCONS control. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

5) Forced alarm messages **`NAS0110I, NAS0111W, NAS0112I`** or **`NAS0130W`** are now generated, respectively, when the SHOW ON, SHOW OFF, SHOW ERR or SHOW MORE|LESS commands are entered to inform the SYSCONS operator of changes to SYSCONS display output. Please refer to the Alert/Alarm Message and Clear/Reset Code Enhancements area of this section for a description of the messages.

SMSG Command - Send Message to Another Console

(changed for V2R4M0)

for a specific console

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	SMSG	**[{LCLCONS|*rconname*}]**	**A**	**240**
		'*msgtext*'	**N**	**114**

or for a collection of consoles

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**SMSG**	**'*msgtext*'**	**N**	**114**

This command is used to send a one line message supplied by the ***msgtext*** string to the HNAS console operator identified by the **ID=** modifier.

- When the **LCLCONS** or ***rconname*** argument is provided, it directs SMSG processing to the identified console. This allows one console to send a message to another console. For example, SMSG RCON0031 'are you there?' will display the text 'are you there?' on the remote console named RCON0031. For more information on LCLCONS|*rconname* processing, please refer to section entitled 'Console PCE name as a Suffix Command Modifier' on page CONS-11.

- When **ID=*minid*[-*maxid*]** is specified, the message is sent to the selected console PCEs. If **ID=0** is specified, the message is broadcast to all active consoles PCEs.

- The ***msgtext*** string must be enclosed in quotes and may be up to 90 characters in length (excluding the quotes).

  Please refer to the Process Control Element Identification (ID=) Table in this manual for additional information concerning ID= assignment.

SNAP Command - Dump HNAS Memory Areas To SYSPRINT (Privileged)

(new for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	SNAP	**[{ALL|*trgtlist*}]**	**A**	**240**

This command is used to dump **ALL** or a specific list (***trgtlist***) of HNAS memory areas.

```
trgtlist={ALOG|ALARMLOG}    <- alarm log
         {BFR|BFRSTOR}      <- buffer pool
         {CNFG|CONFIG}      <- configuration area
         {CVT}              <- communications vector table
         {DNAS|CONSDNAS}    <- DNAS module (NASIDENT)
         {MCH|MCHSTOR}      <- MCH area (MCH,VCB,LUB)
         {MLOG|MAINTLOG}    <- maintenance log
         {PCE|PCESTOR}      <- dynamic PCE area
         {TINT|TCPINTBL}    <- TCPIP interrupt table
         {TRC|TRACE}        <- internal trace table
         {VINT|VTAMINTBL}   <- VTAM interrupt table
         {WA|XFNASWA}       <- workarea
         {WAX}              <- workarea extension
```

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

SNAP Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
WA|XFNASWA	**none**	**1**
WAX	**none**	**2**
CVT	**none**	**3**
MLOG|MAINTLOG	**none**	**4**
ALOG|ALARMLOG	**none**	**5**
DNAS|CONSDNAS	**none**	**6**
TINT|TCPINTBL	**none**	**7**
VINT|VTAMINTBL	**none**	**8**
PCE|PCESTOR	**none**	**9**
BFR|BFRSTOR	**none**	**10**
MCH|MCHSTOR	**none**	**11**
CNFG|CONFIG	**none**	**12**
TRC|TRACE	**none**	**13**

Warning: HNAS SNAP dumping requires additional computing cycles which can influence HNAS performance.

General notes for SNAP:

1) Enter **ALL** to dump all HNAS control areas (ALL is the default when no arguments are specified).

2) Enter a list of areas to be dumped. For example: SNAP WA WAX TINT will dump these 3 areas only.

3) SNAP processing requires additional CPU cycles. HNAS processing is momentarily interrupted while the SNAP dump is taken.

STATS Command - Statistics Collection Control (Privileged)

(changed for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	STATS	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[{ON|OFF}]	**C**	**230**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**STATS**	**[*rmtname*]**	**C**	**230**
		[{ON|OFF}]	**C**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**STATS**	**[{ON|OFF}]**	**C**	**230**

or for the system aggregate

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
ID=0 **RNM=**	**STATS**	**[{ON|OFF}]** **[{ALLTYPES|NOTYPES|** **CONS|NETV|TCP|TMR|** **UTIL|XOT|XTP}]**	**C** **A**	**230** **230**

This command is used to start (**ON**) or stop (**OFF**) *local* PCE statistics collecting for the PCEs identified by the command modifiers (ID= or RNM=), to start (**ID=0 RNM= STATS ON**) or stop (**ID=0 RNM= STATS OFF**) *global* PCE statistics collecting (system aggregate) and/or to identify the type(s) of processes (**ALLTYPES|CONS|NETV|TCP|TMR|UTIL|XOT|XTP**) for which *global* PCE statistics collecting is applicable.

Note: All STATS arguments can be entered at the same time with each one separated by space character. For example, STATS ON TCP TMR. An error message will be generated if duplicate or conflicting arguments are entered. For example, STATS ON ON is rejected because ON is duplicated and STATS ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs STATS processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled 'PCE name as a Suffix Command Modifier' on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting STATS ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, STATS ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, STATS ON|OFF processing is for all PCEs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

  STATS Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
ON	**OFF**	***pcestatic*|*pcename*|RNM=|ID=**	**1**
OFF	**ON**	***pcestatic*|*pcename*|RNM=|ID=**	**2**
ALLTYPES	**NOTYPES**	**system wide**	**3**
NOTYPES	**ALLTYPES**	**system wide**	**4**
CONS	**none**	**system wide**	**5**
NETV	**none**	**system wide**	**6**
TCP	**none**	**system wide**	**7**
TMR	**none**	**system wide**	**8**
UTIL	**none**	**system wide**	**9**
XOT	**none**	**system wide**	**10**
XTP	**none**	**system wide**	**11**

- Enter **STATS [ON]** to start *local* PCE statistics collecting for the PCEs identified by the command modifiers. *Global* PCE statistics collecting will be activated if **ID=0** and **RNM=** (null) is specified. *Global* PCE statistics collecting is provided only for those PCEs that pass the type filter.

- Enter **STATS OFF** to stop *local* PCE statistics collecting for the PCEs identified by the command modifiers. *Global* PCE statistics collecting will be deactivated if **ID=0** and **RNM=** (null) is specified.

- Enter **STATS ALLTYPES** to set all PCE types for *global* statistics collecting.

- Enter **STATS NOTYPES** to reset all PCE types for *global* statistics collecting. This prevents PCE statistics from being collected when only *global* statistics collecting is active (type filter-

ing does not affect *local* statistics collecting).

- Enter **STATS {CONS|NETV|TCP|TMR|UTIL|XOT|XTP}** to set one or more PCE types for *global* statistics collecting.

  Note: To restrict *global* PCE statistics collecting to a specific process or processes, for example TCP and XOT, enter **STATS NOTYPES STATS TCP XOT**.

- To summarize, you may enter a STATS parameter that requests a specific action to be performed as follows:

STATS Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* PCE statistics collecting for the PCEs identified by the ID= or RNM= modifiers.**
OFF	**Deactivate *local* PCE statistics collecting for the PCEs identified by the ID= or RNM= modifiers.**
ID=0 RNM= ON	**Activate g*lobal* PCE statistics collecting (system wide).**
ID=0 RNM= OFF	**Deactivate *global* PCE statistics collecting (system wide).**
ALLTYPES	**Allow *global* PCE statistics collecting for all PCE types.**
NOTYPES	**Inhibit *global* PCE statistics collecting for all PCE types.**
CONS	**Allow *global* PCE statistics collecting for CONSOLE PCEs.**
NETV	**Allow *global* PCE statistics collecting for the NETVIEW PCE (future support).**
TCP	**Allow *global* PCE statistics collecting for TCP/IP PCEs.**
TMR	**Allow *global* PCE statistics collecting for the system TIMER PCE.**
UTIL	**Allow *global* PCE statistics collecting for the system UTILITY PCE.**
XOT	**Allow *global* PCE statistics collecting for the XOT UTILITY PCE.**
XTP	**Allow *global* PCE statistics collecting for the XTP UTILITY PCE.**

Example #1: To start *global* PCE statistics collecting for all PCEs, enter

```
ID=0 RNM= STATS [ON]
```

Example #2: To start *local* PCE statistics collecting for a specific PCE when you are not sure of the current state of statistics collecting within HNAS, enter

```
ID=0 RNM=      <- prepare for global processing
STATS OFF      <- stop global statistics collecting
ID=pceid       <- identify PCE
```

```
STATS [ON]     <- start local statistics collecting for PCE pceid
```

Example #3: To start *local* PCE statistics collecting for all the PCEs associated with a TYPE=XOT REMOTE when you are not sure of the current state of statistics collecting within HNAS, enter

```
ID=0 RNM=      <- prepare for global processing
STATS OFF      <- stop global statistics collecting
RNM=rmtname    <- identify REMOTE
STATS [ON]     <- start local statistics collecting for PCEs on rmtname
```

Warning: HNAS statistics collecting requires additional computing cycles which can influence HNAS performance.

Refer to the description of the DSTATS command on page CONS-123 of this document for information on how to display statistics information.

General notes for STATS:

1) If ID=0 RNM= STATS ALLON and STATS NOTYPES is in effect, no *global* PCE statistics collecting will be performed. Note that the type filter does not affect *local* PCE statistics collecting.

2) The STATS ON|OFF commands do not alter the PCE type(s) that are currently in effect.

3) HNAS activates with **STATS OFF** and **STATS NOTYPES** set in the absence of the **STATS** start parameter.

4) The **STATS [ON]** start parameter performs the same function as the **ID=0 RNM= STATS [ON]** and **STATS ALLTYPES** console commands for *global* PCE statistics collecting. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

5) ON is assumed if STATS is entered with no argument.

6) If **NOTYPES** is in effect when **ON** is set, **ALLTYPES** is forced.

TRCADDR Command - Specific Resource Trace Control (Privileged)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[CLDADDR=*ddd...ddd*]	**TRCADDR**	**[{ADD|DEL|PRG}]**	**N**	**114**
[CLGADDR=*ddd...ddd*]				
[IPADDR=*aaa.bbb.ccc.ddd*[(*ppppp*)]]				

This command is used to add (**ADD**) or remove (**DEL**) a resource address from the HNAS internal resource address table or to purge (**PRG**) the table of all addresses effectively stopping all resource address event tracing.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCADDR Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ADD	**DEL**	**1**
DEL	**ADD**	**2**
PRG	**none**	**3**

- The **TRCADDR** command can be used in lieu of **TRCVC** or **TRCLU** to limit tracing to specific resources. Using the **TRCADDR** command, you can trace events for up to 16 ***called*** DTE addresses and/or 16 ***calling*** DTE addresses and/or 16 **TCP sockets**.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of all HNAS trace entries can be found in the TRACE or CONSTRC book in section Trace.

TRCALL Command - Trace ALL Control (Privileged)

(changed for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCALL	**[{{ON|START|STRT}| {OFF|STOP}] [{SUSP|SUSPEND}| {RSME|RESUME}}]**	**C**	**230**

This command is used to start (**ON|START|STRT**) or stop (**OFF|STOP**) *global* event tracing for **all** resources (PCE, MCH, MCHX, LU and VC), to suspend (**SUSP|SUSPEND**) or to resume (**RSME|RESUME**) active tracing without altering the HNAS trace state.

Note: All TRCALL arguments can be entered at the same time with each one separated by space character. For example, TRCALL ON RSME. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCALL RSME RSME is rejected because RSME is duplicated and TRCALL RSME SUSP is rejected because RSME and SUSP conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCALL Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON|START|STRT	**OFF|STOP**	**1**
OFF|STOP	**ON|START|STRT**	**2**
SUSP|SUSPEND	**RSME|RESUME**	**3**
RSME|RESUME	**SUSP|SUSPEND**	**4**

- Enter **TRCALL [ON]** to start *global* event tracing for all resources (equivalent to the TRCALL start parameter).

Note: TRCALL [ON] is treated as though TRCBFR ALLON, TRCBFR ALLTYPES, TRCDATA ALLON, TRCDATA ALLTYPES, TRCDBK ON, TRCDISP ALLON, TRCDISP ALLTYPES, TRCIO ALLON, TRCIO ALLTYPES, TRCMCH ALLON, TRCMCHX ALLON, TRCLU ALLON and TRCVC ALLON were entered.

Note: Since 230, TRCALL [ON] no longer includes TRCSUBR ON.

- Enter **TRCALL OFF** to stop *global* event tracing for all resources.

  Note: TRCALL OFF is treated as though TRCBFR ALLOFF, TRCBFR NOTYPES, TRCDATA ALLOFF, TRCDATA NOTYPES, TRCDBK OFF, TRCDISP ALLOFF, TRCDISP NOTYPES, TRCIO ALLOFF, TRCIO NOTYPES, TRCMCH ALLOFF, TRCMCHX ALLOFF, TRCLU ALLOFF and TRCVC ALLOFF were entered.

  Note: Since 230, TRCALL OFF no longer includes TRCSUBR OFF.

- Enter **TRCALL SUSP** to manually suspend active tracing.

- Enter **TRCALL RSME** to resume tracing that has been suspended.

- To summarize, you may enter a TRCALL parameter that requests a specific action to be performed as follows:

  TRCALL Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *global* event tracing for all resources.**
OFF	**Deactivate *global* event tracing for all resources.**
SUSP	**Suspend all tracing.**
RSME	**Resume suspended tracing.**
START	**Equivalent to ON.**
STRT	**Equivalent to ON.**
STOP	**Equivalent to OFF.**
SUSPEND	**Equivalent to SUSP.**
RESUME	**Equivalent to RSME.**

  Warning: This console command can start several traces. HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of all HNAS trace entries can be found in the Trace Table Entries section of this product documentation.

  General notes for TRCALL:

  1) HNAS activates with **TRCALL OFF** set in the absence of the **TRCALL** start parameter.

  2) The **TRCALL** start parameter performs the same function as the **TRCALL [ON]**, **TRCLU MAXDATA** and **TRCVC MAXDATA** console commands for *global* event tracing. For

more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCBFR Command - PCE I/O Buffer Trace Control (Privileged)

(changed for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCBFR	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[{ON|OFF}]	**C**	**230**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**TRCBFR**	**[*rmtname*]**	**C**	**230**
		[{ON|OFF}]	**C**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCBFR**	**[{ON|OFF}]**	**C**	**230**

or for all PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCBFR	**[{ALLON|ALLOFF}]** **[{ALLTYPES|NOTYPES|** **CONS|NETV|TCP|TMR|** **UTIL|XOT|XTP}]**	**A** **A**	**230** **230**

This command is used to start (**ON**) or stop (**OFF**) *local* PCE I/O buffer tracing for the PCEs identified by the command modifiers (ID= or RNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* PCE I/O buffer tracing and/or to identify the type(s) of processes (**ALLTYPES|CONS|NETV|TCP|TMR|UTIL|XOT|XTP**) for which *global* PCE I/O buffer tracing is applicable. PCE I/O buffer trace entries are logged only if PCE I/O tracing is also active (see **TRCIO**).

Note: All TRCBFR arguments can be entered at the same time with each one separated by space character. For example, TRCBFR ON TCP TMR. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCBFR ON ON is rejected because ON is duplicated and TRCBFR ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs TRCBFR processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled 'PCE name as a Suffix Command Modifier' on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting TRCBFR ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, TRCBFR ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, TRCBFR ON|OFF processing is for all PCEs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCBFR Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
ON	**OFF**	***pcestatic*|*pcename*|RNM=|ID=**	**1**
OFF	**ON**	***pcestatic*|*pcename*|RNM=|ID=**	**2**
ALLON	**ALLOFF**	**system wide**	**3**
ALLOFF	**ALLON**	**system wide**	**4**
ALLTYPES	**NOTYPES**	**system wide**	**5**
NOTYPES	**ALLTYPES**	**system wide**	**6**
CONS	**none**	**system wide**	**7**
NETV	**none**	**system wide**	**8**
TCP	**none**	**system wide**	**9**
TMR	**none**	**system wide**	**10**
UTIL	**none**	**system wide**	**11**
XOT	**none**	**system wide**	**12**
XTP	**none**	**system wide**	**13**

- Enter **TRCBFR [ON]** to start *local* PCE I/O buffer tracing for the PCEs identified by the command modifiers. *Local* PCE I/O buffer tracing will be activated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **TRCBFR OFF** to stop *local* PCE I/O buffer tracing for the PCEs identified by the command modifiers. *Local* PCE I/O buffer tracing will be deactivated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **TRCBFR ALLON** to start *global* PCE I/O buffer tracing (equivalent to the TRCBFR start parameter). *Global* PCE I/O buffer trace entries are logged only for those PCEs that pass

the type filter.

- Enter **TRCBFR ALLOFF** to stop *global* and all *local* PCE I/O buffer tracing.
- Enter **TRCBFR ALLTYPES** to set all PCE types for *global* PCE I/O buffer tracing.
- Enter **TRCBFR NOTYPES** to reset all PCE types for *global* PCE I/O buffer tracing. This prevents PCE I/O buffer trace entries from being logged when only *global* PCE I/O buffer tracing is active (type filtering does not affect *local* PCE I/O buffer tracing).
- Enter **TRCBFR {CONS|NETV|TCP|TMR|UTIL|XOT|XTP}** to set one or more PCE types for *global* PCE I/O buffer tracing.

  Note: To restrict *global* PCE I/O buffer tracing to a specific process or processes, for example TCP and XOT, enter **TRCBFR NOTYPES TRCBFR TCP XOT**.
- To summarize, you may enter a TRCBFR parameter that requests a specific action to be performed as follows:

TRCBFR Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* PCE I/O buffer tracing for the PCEs identified by the ID= or RNM= modifiers.**
OFF	**Deactivate *local* PCE I/O buffer tracing for the PCEs identified by the ID= or RNM= modifiers.**
ALLON	**Activate *global* PCE I/O buffer tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* PCE I/O buffer tracing (system wide).**
ALLTYPES	**Allow *global* PCE I/O buffer tracing for all PCE types.**
NOTYPES	**Inhibit *global* PCE I/O buffer tracing for all PCE types.**
CONS	**Allow *global* PCE I/O buffer tracing for CONSOLE PCEs.**
NETV	**Allow *global* PCE I/O buffer tracing for the NETVIEW PCE (future support).**
TCP	**Allow *global* PCE I/O buffer tracing for TCP/IP PCEs.**
TMR	**Allow *global* PCE I/O buffer tracing for the system TIMER PCE.**
UTIL	**Allow *global* PCE I/O buffer tracing for the system UTILITY PCE.**
XOT	**Allow *global* PCE I/O buffer tracing for the XOT UTILITY PCE.**
XTP	**Allow *global* PCE I/O buffer tracing for the XTP UTILITY PCE.**

Example #1: To start *global* PCE I/O and PCE I/O buffer tracing for all PCEs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                    <- stop all tracing
```

TRCIO ALLON TRCIO ALLTYPES <- start *global* PCE I/O tracing
TRCBFR ALLON TRCBFR ALLTYPES <- start *global* PCE I/O buffer tracing

Example #2: To start *local* PCE I/O and PCE I/O buffer tracing for a specific PCE when you are not sure of the current state of tracing within HNAS, enter

TRCALL STOP <- stop all tracing
ID=*pceid* <- identify PCE
TRCIO [ON] <- start *local* I/O tracing for PCE *pceid*
TRCBFR [ON] <- start *local* I/O buffer tracing for PCE *pceid*

Example #3: To start *local* PCE I/O and PCE I/O buffer tracing for all the PCEs associated with a TYPE=XOT REMOTE when you are not sure of the current state of tracing within HNAS, enter

TRCALL STOP <- stop all tracing
RNM=*rmtname* <- identify REMOTE
TRCIO [ON] <- start *local* I/O tracing for PCEs on *rmtname*
TRCBFR [ON] <- start *local* I/O buffer tracing for PCEs on *rmtname*

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS buffer trace entries (ID=5|6) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCBFR:

1) If TRCBFR is locally active for a PCE, PCE I/O buffer trace entries will only be logged if TRCIO is also locally active for the PCE or is globally active for the system (TRCIO ALLON).

2) If TRCBFR ALLON and TRCBFR NOTYPES are in effect, no *global* PCE I/O buffer trace entries will be logged. Note that the type filter does not affect *local* PCE I/O buffer tracing.

3) The TRCBFR ON|OFF|ALLON|ALLOFF commands do not alter the PCE type(s) that are currently in effect.

4) HNAS activates with **TRCBFR OFF** and **TRCBFR NOTYPES** set in the absence of the **TRCBFR** start parameter.

5) The **TRCBFR [ON]** start parameter performs the same function as the **TRCBFR ALLON** and **TRCBFR ALLTYPES** console commands for *global* PCE I/O buffer tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

6) ON is assumed if TRCBFR is entered with no argument.

7) If **NOTYPES** is in effect when **ON|ALLON** is set, **ALLTYPES** is forced.

TRCBFRQ Command - XFBFR Call Trace Control (Privileged)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCBFRQ	**[{ON|OFF}]**		

This command is used to start (**ON**) or stop (**OFF**) XFBFR subroutine call tracing.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCBFRQ Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**

- XFBFR subroutine call tracing allows buffer lease and release requests to be logged in the HNAS internal trace table. This functionality is valuable for remote problem diagnosis when the trace table can be examined by a remote console operator.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of HNAS subroutine call trace entries (ID=8) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCBFRQ:

1) HNAS activates with **TRCBFRQ OFF** set in the absence of the **TRCBFRQ** start parameter.

2) The **TRCBFRQ [ON]** and **TRCBFRQ OFF** start parameters perform the same function as the **TRCBFRQ [ON]** and **TRCBFRQ OFF** console commands, respectively, for *global* buffer request tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCBST Command - Binary Search Trace Control (Privileged)

(new for V2R2M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCBST	[{**ON**|**OFF**}]	**N**	**220**

This command is used to start (**ON**) or stop (**OFF**) binary search table update tracing.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCBST Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**

- Binary search table update tracing allows add, delete and search requests to be logged in the HNAS internal trace table. This functionality is valuable for remote problem diagnosis when the trace table can be examined by a remote console operator.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of HNAS binary search trace entries (ID=81) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCBST:

1) HNAS activates with **TRCBST OFF** set in the absence of the **TRCBST** start parameter.

2) The **TRCBST [ON]** and **TRCBST OFF** start parameters perform the same function as the **TRCBST [ON]** and **TRCBST OFF** console commands, respectively, for *global* binary search table tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCCNFG Command - Configuration Call Trace Control (Privileged)

(new for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCCNFG	**[{ON|OFF}]**	**N**	**230**

This command is used to start (**ON**) or stop (**OFF**) configuration subroutine call tracing.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCCNFG Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**

- Configuration subroutine call tracing allows the configuration update process using the MLCL and MRMT console commands to be monitored and logged in the HNAS internal trace table. This functionality is valuable for remote problem diagnosis when the trace table can be examined by a remote console operator.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of HNAS subroutine call trace entries (ID=8) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCCNFG:

1) HNAS activates with **TRCCNFG OFF** set in the absence of the **TRCCNFG** start parameter.

2) The **TRCCNFG [ON]** and **TRCCNFG OFF** start parameters perform the same function as the **TRCCNFG [ON]** and **TRCCNFG OFF** console commands, respectively, for *global* configuration process tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCCONS Command - Console Trace Control (Privileged)

(added for V2R3M0)
(changed for V2R4M0)

for a specific console PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCCONS	**[{LCLCONS|*rconname*}]**	**A**	**240**
		[{ON|OFF}]	**A**	**230**

or for a collection of console PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCCONS**	**[{ON|OFF}]**	**A**	**230**

or for all console PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCCONS	**[{ALLON|ALLOFF}]**	**A**	**230**

This command is used to start (**ON**) or stop (**OFF**) *local* console PCE event tracing for the PCEs identified by the ID= command modifiers or to start (**ALLON**) or stop (**ALLOFF**) *global* console PCE tracing.

Console event tracing allows console activity to be monitored and logged in the HNAS internal trace table. This functionality is valuable for remote problem diagnosis when the trace table can be examined by a remote console operator.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the **LCLCONS** or ***rconname*** argument is provided, it directs EXEC processing to the identified console. This allows one console to alter TRCCONS processing for another console. For example, TRCCONS RCON0031 ON will cause the console tracing to be started for the remote console named RCON0031 regardless of which console actually entered the command. For more information on LCLCONS|*rconname* processing, please refer to section entitled ‘Console PCE name as a Suffix Command Modifier’ on page CONS-11.

- When **ID=*minid*[-*maxid*]** is specified, TRCCONS ON|OFF processing is restricted to the selected console PCEs. If **ID=0** is specified, TRCCONS ON|OFF processing is for all console PCEs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCCONS Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
ON	**OFF**	**LCLCONS|*rconname*|ID=**	**1**
OFF	**ON**	**LCLCONS|*rconname*|ID=**	**2**
ALLON	**ALLOFF**	**system wide**	**3**
ALLOFF	**ALLON**	**system wide**	**4**

- Enter **TRCCONS [ON]** to start *local* console PCE event tracing for the PCEs identified by the command modifiers. *Local* console PCE event tracing will be activated for all console PCEs if **ID=0** is specified.
- Enter **TRCCONS OFF** to stop *local* console PCE event tracing for the PCEs identified by the command modifiers. *Local* console PCE event tracing will be deactivated for all console PCEs if **ID=0** is specified.
- Enter **TRCCONS ALLON** to start *global* console event tracing (equivalent to the TRCCONS start parameter).
- Enter **TRCCONS ALLOFF** to stop *global* and all *local* console PCE event tracing.
- To summarize, you may enter a TRCCONS parameter that requests a specific action to be performed as follows:

TRCCONS Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* console PCE event tracing for the PCEs identified by the ID= modifier.**
OFF	**Deactivate *local* console PCE event tracing for the PCEs identified by the ID= modifier.**
ALLON	**Activate *global* console PCE event tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* console PCE event tracing (system wide).**

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS console event trace entries (ID=9) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCCONS:

1) HNAS activates with **TRCCONS ALLOFF** set in the absence of the **TRCCONS** start parameter.

2) The **TRCCONS [ALLON]** and **TRCCONS ALLOFF** start parameter performs the same function as the **TRCCONS ALLON** and **TRCCONS ALLOFF** console commands, respectively, for *global* console process tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCDATA Command - PCE I/O Data Trace Control (Privileged)

(changed for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCDATA	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[{ON|OFF}]	**C**	**230**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**TRCDATA**	**[*rmtname*]**	**C**	**230**
		[{ON|OFF}]	**C**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCDATA**	**[{ON|OFF]**	**C**	**230**

or for all PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCDATA	**[{ALLON|ALLOFF}]** **[{ALLTYPES|NOTYPES|** **CONS|NETV|TCP|TMR|** **UTIL|XOT|XTP}]**	**A** **A**	**230** **230**

This command is used to start (**ON**) or stop (**OFF**) *local* PCE I/O data tracing for the PCEs identified by the command modifiers (ID= or RNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* PCE I/O data tracing and/or to identify the type(s) of processes (**ALLTYPES|CONS|NETV|TCP|TMR|UTIL|XOT|XTP**) for which *global* PCE I/O data tracing is applicable. PCE I/O data trace entries are logged only if PCE I/O tracing is also active (see **TRCIO**).

Note: All TRCDATA arguments can be entered at the same time with each one separated by space character. For example, TRCDATA ON TCP TMR. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCDATA ON ON is rejected because ON is duplicated and TRCDATA ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs TRCDATA processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled 'PCE name as a Suffix Command Modifier' on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting TRCDATA ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, TRCDATA ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, TRCDATA ON|OFF processing is for all PCEs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCDATA Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
ON	**OFF**	***pcestatic*|*pcename*|RNM=|ID=**	**1**
OFF	**ON**	***pcestatic*|*pcename*|RNM=|ID=**	**2**
ALLON	**ALLOFF**	**system wide**	**3**
ALLOFF	**ALLON**	**system wide**	**4**
ALLTYPES	**NOTYPES**	**system wide**	**5**
NOTYPES	**ALLTYPES**	**system wide**	**6**
CONS	**none**	**system wide**	**7**
NETV	**none**	**system wide**	**8**
TCP	**none**	**system wide**	**9**
TMR	**none**	**system wide**	**10**
UTIL	**none**	**system wide**	**11**
XOT	**none**	**system wide**	**12**
XTP	**none**	**system wide**	**13**

- Enter **TRCDATA [ON]** to start *local* PCE I/O data tracing for the PCEs identified by the command modifiers. *Local* PCE I/O data tracing will be activated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **TRCDATA OFF** to stop *local* PCE I/O data tracing for the PCEs identified by the command modifiers. *Local* PCE I/O data tracing will be deactivated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **TRCDATA ALLON** to start *global* PCE I/O data tracing (equivalent to the TRCDATA start parameter). *Global* PCE I/O data trace entries are logged only for those PCEs that

pass the type filter.

- Enter **TRCDATA ALLOFF** to stop *global* and all *local* PCE I/O data tracing.
- Enter **TRCDATA ALLTYPES** to set all PCE types for *global* PCE I/O data tracing.
- Enter **TRCDATA NOTYPES** to reset all PCE types for *global* PCE I/O data tracing. This prevents PCE I/O data trace entries from being logged when only *global* PCE I/O data tracing is active (type filtering does not affect *local* PCE I/O data tracing).
- Enter **TRCDATA {CONS|NETV|TCP|TMR|UTIL|XOT|XTP}** to set one or more PCE types for *global* PCE I/O data tracing.

  Note: To restrict *global* PCE I/O data tracing to a specific process or processes, for example TCP and XOT, enter **TRCDATA NOTYPES TRCDATA TCP XOT**.
- To summarize, you may enter a TRCDATA parameter that requests a specific action to be performed as follows:

TRCDATA Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* PCE I/O data tracing for the PCEs identified by the ID= or RNM= modifiers.**
OFF	**Deactivate *local* PCE I/O data tracing for the PCEs identified by the ID= or RNM= modifiers.**
ALLON	**Activate *global* PCE I/O data tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* PCE I/O data tracing (system wide).**
ALLTYPES	**Allow *global* PCE I/O data tracing for all PCE types.**
NOTYPES	**Inhibit *global* PCE I/O data tracing for all PCE types.**
CONS	**Allow *global* PCE I/O data tracing for CONSOLE PCEs.**
NETV	**Allow *global* PCE I/O data tracing for the NETVIEW PCE (future support).**
TCP	**Allow *global* PCE I/O data tracing for TCP/IP PCEs.**
TMR	**Allow *global* PCE I/O data tracing for the system TIMER PCE.**
UTIL	**Allow *global* PCE I/O data tracing for the system UTILITY PCE.**
XOT	**Allow *global* PCE I/O data tracing for the XOT UTILITY PCE.**
XTP	**Allow *global* PCE I/O data tracing for the XTP UTILITY PCE.**

Example #1: To start *global* PCE I/O and PCE I/O data tracing for all PCEs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                          <- stop all tracing
TRCIO ALLON TRCIO ALLTYPES           <- start global PCE I/O tracing
TRCDATA ALLON TRCDATA ALLTYPES       <- start global PCE I/O data tracing
```

Example #2: To start *local* PCE I/O and PCE I/O data tracing for a specific PCE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                  <- stop all tracing
ID=pceid                     <- identify PCE
TRCIO [ON]                   <- start local I/O tracing for PCE pceid
TRCDATA [ON]                 <- start local I/O data tracing for PCE pceid
```

Example #3: To start *local* PCE I/O and PCE I/O data tracing for all the PCEs associated with a TYPE=XOT REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                  <- stop all tracing
RNM=rmtname                  <- identify REMOTE
TRCIO [ON]                   <- start local I/O tracing for PCEs on rmtname
TRCDATA [ON]                 <- start local I/O data tracing for PCEs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS data trace entries (ID=7) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCDATA:

1) If TRCDATA is locally active for a PCE, PCE I/O data trace entries will only be logged if TRCIO is also locally active for the PCE or is globally active for the system (TRCIO ALLON).

2) If TRCDATA ALLON and TRCDATA NOTYPES are in effect, no *global* PCE I/O data trace entries will be logged. Note that the type filter does not affect *local* PCE I/O data tracing.

3) The TRCDATA ON|OFF|ALLON|ALLOFF commands do not alter the PCE type(s) that are currently in effect.

4) HNAS activates with **TRCDATA OFF** and **TRCDATA NOTYPES** set in the absence of the **TRCDATA** start parameter.

5) The **TRCDATA [ON]** start parameter performs the same function as the **TRCDATA ALLON** and **TRCDATA ALLTYPES** console commands for *global* PCE I/O data tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

6) ON is assumed if TRCDATA is entered with no argument.

7) If **NOTYPES** is in effect when **ON|ALLON** is set, **ALLTYPES** is forced.

TRCDBK Command - Data Block Trace Control (Privileged)

(new for V2R2M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCDBK	**[{ON|OFF}]**	**N**	**220**

This command is used to start (**ON**) or stop (**OFF**) *global* (common) data block tracing for all LUs and VCs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCDBK Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2

- Data block tracing allows extended control block information to be logged in the HNAS internal trace table or written in the SYSPRINT log (if **PRNT ON** is in effect). This functionality is valuable for remote problem diagnosis when the trace table can be examined by a remote console operator.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  Note: Unless instructed by Comm-Pro technical personnel, the use of TRCDBK should be avoided due to the amount of additional data that can be logged in SYSPRINT.

General notes for TRCDBK:

1) HNAS activates with **TRCDBK OFF** set in the absence of the **TRCDBK** start parameter.

2) The **TRCDBK [ON]** start parameter performs the same function as the **TRCDBK [ON]** console command for *global* data block tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCDISP Command - PCE Dispatcher Trace Control (Privileged)

(changed for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCDISP	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[NQDQ] [{ON|OFF}]	**C**	**230**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**TRCDISP**	**[*rmtname*]**	**C**	**230**
		[NQDQ] [{ON|OFF}]	**C**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCDISP**	**[NQDQ] [{ON|OFF}]**	**C**	**230**

or for all PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCDISP	**[NQDQ] [{ALLON|ALLOFF}]** **[{ALLTYPES|NOTYPES|** **CONS|NETV|TCP|TMR|** **UTIL|XOT|XTP}]**	**A** **A**	**230** **230**

This command is used to start (**ON**) or stop (**OFF**) *local* PCE dispatcher tracing for the PCEs identified by the command modifiers (ID= or RNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* PCE dispatcher tracing and/or to identify the type(s) of processes (**ALLTYPES|CONS|NETV|TCP|TMR|UTIL|XOT|XTP**) for which *global* PCE dispatcher tracing is applicable. If **NQDQ** is entered before ON (*local*) or ALLON (*global*), it causes XFWAIT, XFPOST, XFSTMR and XFRTMR macro calls to be logged with the dispatcher trace entries.

Note: All TRCDISP arguments can be entered at the same time with each one separated by space character. For example, TRCDISP ON TCP TMR. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCDISP ON ON is rejected because ON is duplicated and TRCDISP ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs TRCDISP processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled 'PCE name as a Suffix Command Modifier' on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting TRCDISP ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, TRCDISP ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, TRCDISP ON|OFF processing is for all PCEs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCDISP Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
[NQDQ] ON	**OFF**	***pcestatic*|*pcename*|RNM=|ID=**	**1**
OFF	**ON**	***pcestatic*|*pcename*|RNM=|ID=**	**2**
[NQDQ] ALLON	**ALLOFF**	**system wide**	**3**
ALLOFF	**ALLON**	**system wide**	**4**
ALLTYPES	**NOTYPES**	**system wide**	**5**
NOTYPES	**ALLTYPES**	**system wide**	**6**
CONS	**none**	**system wide**	**7**
NETV	**none**	**system wide**	**8**
TCP	**none**	**system wide**	**9**
TMR	**none**	**system wide**	**10**
UTIL	**none**	**system wide**	**11**
XOT	**none**	**system wide**	**12**
XTP	**none**	**system wide**	**13**

- Enter **TRCDISP [[NQDQ] ON]** to start *local* PCE dispatcher tracing for the PCEs identified by the command modifiers. *Local* PCE dispatcher tracing will be activated for all PCEs if **ID=0** and **RNM=** (null) is specified. When **NQDQ ON** is specified, subtask XFWAIT/XFPOST and XFSTMR/XFRTMR tracing is activated in addition to dispatcher tracing for the identified PCEs.

- Enter **TRCDISP OFF** to stop *local* PCE dispatcher tracing for the PCEs identified by the command modifiers. *Local* PCE dispatcher tracing will be deactivated for all PCEs if **ID=0** and

RNM= (null) is specified.

- Enter **TRCDISP [NQDQ] ALLON** to start *global* PCE dispatcher tracing (equivalent to the TRCDISP start parameter). *Global* PCE dispatcher trace entries are logged only for those PCEs that pass the type filter. When **NQDQ ALLON** is specified, subtask XFWAIT/XFPOST and XFSTMR/XFRTMR tracing is activated in addition to dispatcher tracing for all PCEs.

- Enter **TRCDISP ALLOFF** to stop *global* and all *local* PCE dispatcher tracing.

- Enter **TRCDISP ALLTYPES** to set all PCE types for *global* PCE dispatcher tracing.

- Enter **TRCDISP NOTYPES** to reset all PCE types for *global* PCE dispatcher tracing. This prevents PCE dispatcher trace entries from being logged when only *global* PCE dispatcher tracing is active (type filtering does not affect *local* PCE dispatcher tracing).

- Enter **TRCDISP {CONS|NETV|TCP|TMR|UTIL|XOT|XTP}** to set one or more PCE types for *global* PCE dispatcher tracing.

  Note: To restrict *global* PCE dispatcher tracing to a specific process or processes, for example TCP and XOT, enter **TRCDISP NOTYPES TRCDISP TCP XOT**.

- To summarize, you may enter a TRCDISP parameter that requests a specific action to be performed as follows:

TRCDISP Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* PCE dispatcher tracing for the PCEs identified by the ID= or RNM= modifiers.**
NQDQ ON	**Activate *local* PCE dispatcher tracing plus XFWAIT, XFPOST, XFSTMR and XFRTMR macro tracing for the PCEs identified by the ID= or RNM= modifiers.**
OFF	**Deactivate *local* PCE dispatcher tracing for the PCEs identified by the ID= or RNM= modifiers.**
ALLON	**Activate *global* PCE dispatcher tracing (system wide).**
NQDQ ALLON	**Activate *global* PCE dispatcher tracing plus XFWAIT, XFPOST, XFSTMR and XFRTMR macro tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* PCE dispatcher tracing (system wide).**
ALLTYPES	**Allow *global* PCE dispatcher tracing for all PCE types.**
NOTYPES	**Inhibit *global* PCE dispatcher tracing for all PCE types.**
CONS	**Allow *global* PCE dispatcher tracing for CONSOLE PCEs.**
NETV	**Allow *global* PCE dispatcher tracing for the NETVIEW PCE (future support).**
TCP	**Allow *global* PCE dispatcher tracing for TCP/IP PCEs.**

PARAMETER	ACTION
TMR	**Allow *global* PCE dispatcher tracing for the system TIMER PCE.**
UTIL	**Allow *global* PCE dispatcher tracing for the system UTILITY PCE.**
XOT	**Allow *global* PCE dispatcher tracing for the XOT UTILITY PCE.**
XTP	**Allow *global* PCE dispatcher tracing for the XTP UTILITY PCE.**

Example #1: To start *global* PCE dispatcher tracing for all PCEs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                          <- stop all tracing
TRCDISP ALLON TRCDISP ALLTYPES       <- start global PCE dispatcher tracing
```

Example #2: To start *local* PCE dispatcher tracing for a specific PCE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP              <- stop all tracing
ID=pceid                 <- identify PCE
TRCDISP [ON]             <- start local dispatcher tracing for PCE pceid
```

Example #3: To start *local* PCE dispatcher tracing for all the PCEs associated with a TYPE=XOT REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP              <- stop all tracing
RNM=rmtname              <- identify REMOTE
TRCDISP [ON]             <- start local dispatcher tracing for PCEs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS dispatcher trace entries (ID=0|1|2|9|A|B|C) can be found starting in the Trace Table Entries section of this product documentation.

General notes for TRCDISP:

1) If TRCDISP ALLON and TRCDISP NOTYPES are in effect, no *global* PCE dispatcher trace entries will be logged. Note that the type filter does not affect *local* PCE dispatcher tracing.

2) The TRCDISP ON|OFF|ALLON|ALLOFF commands do not alter the PCE type(s) that are currently in effect.

3) HNAS activates with **TRCDISP OFF** and **TRCDISP NOTYPES** set in the absence of the **TRCDISP** start parameter.

4) The **TRCDISP [ON]** start parameter performs the same function as the **TRCDISP ALLON** and **TRCDISP ALLTYPES** console commands for *global* PCE dispatcher tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

5) ON is assumed if TRCDISP is entered with no argument.

6) If **NOTYPES** is in effect when **ON|ALLON** is set, **ALLTYPES** is forced.

TRCIO Command - PCE I/O Trace Control (Privileged)

(changed for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCIO	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[{ON|OFF}]	**C**	**230**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**TRCIO**	**[*rmtname*]**	**C**	**230**
		[{ON|OFF}]	**C**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCIO**	**[{ON|OFF}}]**	**C**	**230**

or for all PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCIO	**[{ALLON|ALLOFF}]** **[{ALLTYPES|NOTYPES|** **CONS|NETV|TCP|TMR|** **UTIL|XOT|XTP}]**	**A** **A**	**230** **230**

This command is used to start (**ON**) or stop (**OFF**) *local* PCE I/O tracing for the PCEs identified by the command modifiers (ID= or RNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* PCE I/O tracing and/or to identify the type(s) of processes (**ALLTYPES|CONS|NETV|TCP|TMR|UTIL|XOT|XTP**) for which *global* PCE I/O tracing is applicable.

Note: All TRCIO arguments can be entered at the same time with each one separated by space character. For example, TRCIO ON TCP TMR. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCIO ON ON is rejected because ON is duplicated and TRCIO ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs TRCIO processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled 'PCE name as a Suffix Command Modifier' on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting TRCIO ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, TRCIO ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, TRCIO ON|OFF processing is for all PCEs.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCIO Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
ON	**OFF**	***pcestatic*|*pcename*|RNM=|ID=**	**1**
OFF	**ON**	***pcestatic*|*pcename*|RNM=|ID=**	**2**
ALLON	**ALLOFF**	**system wide**	**3**
ALLOFF	**ALLON**	**system wide**	**4**
ALLTYPES	**NOTYPES**	**system wide**	**5**
NOTYPES	**ALLTYPES**	**system wide**	**6**
CONS	**none**	**system wide**	**7**
NETV	**none**	**system wide**	**8**
TCP	**none**	**system wide**	**9**
TMR	**none**	**system wide**	**10**
UTIL	**none**	**system wide**	**11**
XOT	**none**	**system wide**	**12**
XTP	**none**	**system wide**	**13**

- Enter **TRCIO [ON]** to start *local* PCE I/O tracing for the PCEs identified by the command modifiers. *Local* PCE I/O tracing will be activated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **TRCIO OFF** to stop *local* PCE I/O tracing for the PCEs identified by the command modifiers. *Local* PCE I/O tracing will be deactivated for all PCEs if **ID=0** and **RNM=** (null) is specified.

- Enter **TRCIO ALLON** to start *global* PCE I/O tracing (equivalent to the TRCIO start parameter). *Global* PCE I/O trace entries are logged only for those PCEs that pass the type filter.

- Enter **TRCIO ALLOFF** to stop *global* and all *local* PCE I/O tracing.

- Enter **TRCIO ALLTYPES** to set all PCE types for *global* PCE I/O tracing.

- Enter **TRCIO NOTYPES** to reset all PCE types for *global* PCE I/O tracing. This prevents PCE I/O trace entries from being logged when only *global* PCE I/O tracing is active (type filtering does not affect *local* PCE I/O tracing).

- Enter **TRCIO {CONS|NETV|TCP|TMR|UTIL|XOT|XTP}** to set one or more PCE types for *global* PCE I/O tracing.

  Note: To restrict *global* PCE I/O tracing to a specific process or processes, for example TCP and XOT, enter **TRCIO NOTYPES TRCIO TCP XOT**.

- To summarize, you may enter a TRCIO parameter that requests a specific action to be performed as follows:

TRCIO Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* PCE I/O tracing for the PCEs identified by the ID= or RNM= modifiers.**
OFF	**Deactivate *local* PCE I/O tracing for the PCEs identified by the ID= or RNM= modifiers.**
ALLON	**Activate *global* PCE I/O tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* PCE I/O tracing (system wide).**
ALLTYPES	**Allow *global* PCE I/O tracing for all PCE types.**
NOTYPES	**Inhibit *global* PCE I/O tracing for all PCE types.**
CONS	**Allow *global* PCE I/O tracing for CONSOLE PCEs.**
NETV	**Allow *global* PCE I/O tracing for the NETVIEW PCE (future support).**
TCP	**Allow *global* PCE I/O tracing for TCP/IP PCEs.**
TMR	**Allow *global* PCE I/O tracing for the system TIMER PCE.**
UTIL	**Allow *global* PCE I/O tracing for the system UTILITY PCE.**
XOT	**Allow *global* PCE I/O tracing for the XOT UTILITY PCE.**
XTP	**Allow *global* PCE I/O tracing for the XTP UTILITY PCE.**

Example #1: To start *global* PCE I/O tracing for all PCEs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                          <- stop all tracing
TRCIO ALLON TRCIO ALLTYPES           <- start global PCE I/O tracing
```

Example #2: To start *local* PCE I/O tracing for a specific PCE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                          <- stop all tracing
ID=pceid                             <- identify PCE
TRCIO [ON]                           <- start local I/O tracing for PCE pceid
```

Example #3: To start *local* PCE I/O tracing for all the PCEs associated with a TYPE=XOT REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                          <- stop all tracing
RNM=rmtname                          <- identify REMOTE
TRCIO [ON]                           <- start local I/O tracing for PCEs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS I/O trace entries (ID=3|4) can be found starting in the Trace Table Entries section of this product documentation.

General notes for TRCIO:

1) If TRCIO ALLON and TRCIO NOTYPES are in effect, no *global* PCE I/O trace entries will be logged. Note that the type filter does not affect *local* PCE I/O tracing.

2) The TRCIO ON|OFF|ALLON|ALLOFF commands do not alter the PCE type(s) that are currently in effect.

3) HNAS activates with **TRCIO OFF** and **TRCIO NOTYPES** set in the absence of the **TRCIO** start parameter.

4) The **TRCIO [ON]** start parameter performs the same function as the **TRCIO ALLON** and **TRCIO ALLTYPES** console commands for *global* PCE I/O tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

5) ON is assumed if TRCIO is entered with no argument.

6) If **NOTYPES** is in effect when **ON|ALLON** is set, **ALLTYPES** is forced.

TRCLU Command - LU Trace Control (Privileged)

(changed for V2R2M0)

for a specific LU

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[LUNM=*sluname*]	**TRCLU**	**[*sluname*]**	**C**	**230**
		[{ON|OFF}] **[DBK {ON|OFF}]** **[{MAXDATA|MINDATA|NODATA}]**	**C** **C** **A**	**220** **240** **220**

or for the LUs on a specific MCH or SPU

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM={*mchname*|*spuname*}]	**TRCLU**	**[{*mchname*|*spuname*}]**	**C**	**230**
[CID=*mincid*[-*maxcid*]]		**[{ON|OFF}]** **[DBK {ON|OFF}]** **[{MAXDATA|MINDATA|NODATA}]**	**C** **C** **A**	**220** **240** **220**

or for a collection of LUs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCLU**	**[{ON|OFF}]** **[DBK {ON|OFF}]** **[{MAXDATA|MINDATA|NODATA}]**	**C** **C** **A**	**220** **240** **220**
[IFN=*minifn*[-*maxifn*]]				
[CID=*mincid*[-*maxcid*]]				

or for all LUs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCLU	**[{ALLON|ALLOFF}]** **[{MAXDATA|MINDATA|NODATA}]**	**C** **A**	**220** **220**

This command is used to start (**ON**) or stop (**OFF**) *local* LU event tracing or to enable *local* data block logging (**DBK**) for the LUs identified by the command modifiers (ID=, IFN=, CID=, RNM=, LUNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* LU event tracing or to control the amount of data (**MAXDATA**|**MINDATA**|**NODATA**) that is logged for the LUs currently being traced.

Note: All TRCLU arguments can be entered at the same time with each one separated by space character. For example, TRCLU ON MAXDATA. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCLU ON ON is rejected because ON is duplicated and TRCLU ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **LUNM=*sluname*** is specified, it overrides the **ID=**, **IFN=** and **CID=** and **RNM=** values restricting the TRCLU ON|OFF|DBK processing to the named LU only.

  Note: When an **LUNM=** value is specified for the **TRCLU** command, a VC trace is also enabled for the duration of the session (PCNE|GATE|PAD sessions only).

- When **RNM={*mchname*|*spuname*}** is specified, it overrides the **ID=** and **IFN=** values restricting TRCLU ON|OFF|DBK processing to the named MCH or SPU for the specified **CID=** values. TRCLU ON|OFF|DBK processing will affect the LUs on the named MCH only.

- When **CID=*mincid*[-*maxcid*]** is specified, TRCLU ON|OFF|DBK processing is restricted to the selected LUs on the selected MCHs. If **CID=0** is specified, TRCLU ON|OFF|DBK processing is for all LUs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), TRCLU ON|OFF|DBK processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, TRCLU ON|OFF|DBK processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, TRCLU ON|OFF|DBK processing is restricted to the selected PCEs. If **ID=0** is specified, TRCLU ON|OFF|DBK processing is for all PCEs. *The ID= modifier is only used if the RNM= and LUNM= modifiers are not set. If ID= is also null, the command is rejected.*

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCLU Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLON	ALLOFF	3
ALLOFF	ALLON	4
DBK ON	DBK OFF	5
DBK OFF	DBK ON	6
MAXDATA	MINDATA|NODATA	7
MINDATA	MAXDATA|NODATA	8
NODATA	MINDATA|MAXDATA	9

-

- Enter **TRCLU [ON]** to start *local* LU event tracing for the LUs identified by the command modifiers.

- Enter **TRCLU OFF** to stop *local* LU event tracing and *local* data block logging (disables DBK) for the LUs identified by the command modifiers.

- Enter **TRCLU DBK [ON]** to enable *local* data block logging for the LUs identified by the command modifiers. This is equivalent to MAXDATA on an LU basis rather than a system wide basis.

- Enter **TRCLU DBK OFF** to disable *local* data block logging for the LUs identified by the command modifiers. This is equivalent to MAXDATA on an LU basis rather than a system wide basis.

- Enter **TRCLU ALLON** to start *global* LU event tracing (equivalent to the TRCLU start parameter).

- Enter **TRCLU ALLOFF** to stop *global* and *local* LU event tracing and *local* data block logging (disables DBK) for all defined LUs.

- Enter **TRCLU MAXDATA** to enable *global* data block logging with a maximum amount of data being logged for the LUs currently being traced.

- Enter **TRCLU MINDATA** to enable *global* data block logging with a minimum amount of data being logged for the LUs currently being traced.

- Enter **TRCLU NODATA** to disable *global* data block logging for the LUs currently being traced. This prevents any additional data being logged when an LU event trace entry is recorded.

  Note: For HNAS release V2R2M0, the MAXDATA, MINDATA and NODATA parameters also started event tracing for all LUs defined to HNAS. This precluded their use to control data logging for a single LU. For this reason, the LU event start function has been removed from these parameters. *Global* LU event tracing must now be started and stopped using the ALLON and ALLOFF parameters, respectively.

- To summarize, you may enter a TRCLU parameter that requests a specific action to be performed as follows:

  TRCLU Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* LU event tracing for the LUs identified by the ID=, IFN= and CID= modifiers or the RNM= and CID= modifiers or the LUNM= modifier.**
OFF	**Deactivate *local* LU event tracing and *local* data block logging for the LUs identified by the ID=, IFN= and CID= modifiers or the RNM= and CID= modifiers or the LUNM= modifier.**

PARAMETER	ACTION
DBK ON or omitted	**Enable *local* data block logging for the LUs identified by the ID=, IFN= and CID= modifiers or the RNM= and CID= modifiers or the LUNM= modifier.**
DBK OFF	**Disable *local* data block logging for the LUs identified by the ID=, IFN= and CID= modifiers or the RNM= and CID= modifiers or the LUNM= modifier.**
ALLON	**Activate *global* LU event tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* LU event tracing (system wide).**
MAXDATA	**Activate *global* data block logging with a maximum amount of data being logged for the LUs currently being traced.**
MINDATA	**Activate *global* data block logging with a minimum amount of data being logged for the LUs currently being traced.**
NODATA	**Deactivate *global* data block logging for the LUs currently being traced.**

Example #1: To start *global* LU event tracing for all LUs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
TRCLU ALLON          <- start global LU event tracing
```

Example #2: To start *local* LU event tracing for a specific LU when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
LUNM=sluname         <- identify LU
TRCLU [ON]           <- start local event tracing for LU sluname
```

Example #3: To start *local* LU event tracing for all the LUs associated with a TYPE=MCH|SPU REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
RNM=rmtname          <- identify REMOTE
TRCLU [ON]           <- start local event tracing for LUs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS LU event trace entries (ID=VOPN|VCLS|...) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCLU:

1) Entering **TRCLU OFF** will terminate *local* event tracing and *local* data block (**DBK**) logging.

2) Entering **TRCLU ON|OFF|DBK|ALLON|ALLOFF** will not alter the *global* data block logging state (**MAXDATA|MINDATA|NODATA**).

3) The **TRCLU** command now produces expanded trace entries containing formatted inbound and outbound call request packet fields. The expanded information is located in the SYSPRINT output (not SYSCONS) and is only generated when **TRCLU** is enabled or when **TRCMCH ICR** or **OCR** options are enabled. Please refer to **TRCMCH** command description to view sample **NASnnnnT** trace alert message layouts.

4) HNAS activates with **TRCLU ON** and **TRCLU MINDATA** set in the absence of the **TRCLU** start parameter.

5) The **TRCLU [ON]**, **TRCLU OFF**, **TRCLU MAXDATA**, **TRCLU MINDATA** and **TRCLU NODATA** start parameters perform the same functions as the **TRCLU ALLON**, **TRCLU ALLOFF**, **TRCLU MAXDATA**, **TRCLU MINDATA** and **TRCLU NODATA** console commands, respectively, for *global* LU event tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCLUQ Command - MCH LU Queue Trace Control (Privileged)

for the LUQs on a specific MCH

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*mchname*]	**TRCLUQ**	**[*mchname*]**	**C**	**230**
[LUN=*minlun*[-*maxlun*]]		**[{ON|OFF}]** **[DBK {ON|OFF}]**	C	240

or for a collection of LUQs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCLUQ**	**[{ON|OFF}]** **[DBK {ON|OFF}]**	C	240
[IFN=*minifn*[-*maxifn*]]				
[LUN=*minlun*[-*maxlun*]]				

or for all LUQs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCLUQ	**[{ALLON|ALLOFF}]**	**C**	**230**

This command is used to start (**ON**) or stop (**OFF**) *local* LU event tracing or to initiate *local* data block logging (**DBK**) as new LU connections are established for the MCHs and MCHXs identified by the command modifiers (ID=, IFN=, LUN=, RNM=). *Local* LU event tracing and *local* data block logging for **all** LUQs can be started (**ALLON**) and stopped (**ALLOFF**). This command is provided so that LU tracing can be initiated as soon as a LUB is added to its active LU queue.

Note: All TRCLUQ arguments can be entered at the same time with each one separated by space character. For example, TRCLUQ ON DBK ON. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCLUQ ON ON is rejected because ON is duplicated and TRCLUQ ON OFF is rejected because ON and OFF conflict.

Note: LUQ=MCHLUQ for LLC0 (PCNE) and LLC5 (PAD) LUs (contained in MCH) while LUQ=MCHXLUAQ for LLC4 (GATE) LUs (contained in MCHX).

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting TRCLUQ processing to the named MCH only for the specified **LUN=** values. TRCLUQ processing will affect the LUQs on the named MCH only.

- When **LUN=*minlun*[-*maxlun*]** is specified, TRCLUQ processing is restricted to the selected MCHXs on the selected MCHs. If **LUN=0** is specified, TRCLUQ processing is for all MCHXs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), TRCLUQ processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, TRCLUQ processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, TRCLUQ processing is restricted to the selected PCEs. If **ID=0** is specified, TRCLUQ processing is for all PCEs. *The ID= modifier is only used if the RNM= modifier is not set. If ID= is also null, the command is rejected.*

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCLUQ Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLON	ALLOFF	3
ALLOFF	ALLON	4
DBK ON	DBK OFF	5
DBK OFF	DBK ON	6

- Enter **TRCLUQ [ON]** to start *local* LUQ event tracing for the LUQs identified by the command modifiers.

- Enter **TRCLUQ OFF** to stop *local* LUQ event tracing and *local* data block logging (disables DBK) for the LUQs identified by the command modifiers.

- Enter **TRCLUQ DBK [ON]** to enable *local* data block logging for the LUQs identified by the command modifiers.

- Enter **TRCLUQ DBK OFF** to disable *local* data block logging for the LUQs identified by the command modifiers.

- Enter **TRCLUQ ALLON** to start *local* LUQ event tracing and *local* data block logging (enables DBK) for **all** LUQs.

- Enter **TRCLUQ ALLOFF** to stop *local* LUQ event tracing and *local* data block logging (disables DBK) for **all** LUQs.

- To summarize, you may enter a TRCLUQ parameter that requests a specific action to be performed as follows:

TRCLUQ Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* LUQ event tracing for the LUQs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers.**
OFF	**Deactivate *local* LUQ event tracing and *local* data block logging for the LUQs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers.**
DBK ON or omitted	**Enable *local* data block logging for the LUQs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers.**
DBK OFF	**Disable *local* data block logging for the LUQs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers.**
ALLON	**Activate all *local* LUQ event tracing and *local* data block logging (system wide).**
ALLOFF	**Deactivate all *local* LUQ event tracing and *local* data block logging (system wide).**

Example #1: To start *local* LUQ event tracing for all LUQs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
ID=0 IFN=0 LUN=0            <- force all LUQs to be referenced
TRCLUQ [ON]                 <- start local event tracing for all LUQs
```

Example #2: To start *local* LUQ event tracing for all the LUQs associated with a TYPE=MCH|XTP REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
RNM=rmtname                 <- identify REMOTE
LUN=0                       <- operate on MCH and all MCHXs
TRCLUQ [ON]                 <- start local event tracing for LUQs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS LU event trace entries (ID=VOPN|VCLS|...) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCLUQ:

1) HNAS activates with **TRCLUQ OFF** set. There is no equivalent start parameter.

TRCMCH Command - MCH Trace Control (Privileged)

for a specific MCH

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
`[RNM=mchname]`	`TRCMCH`	`[mchname]`	`C`	`230`

		`[{ON|OFF}]`
		`[{LCL|GBL} [ICR {ON|OFF}]`
		`[ICRF {ON|OFF}]`
		`[OCR {ON|OFF}]`
		`[ICLR {ON|OFF}]`
		`[OCLR {ON|OFF}]]`
			`C`
			`A`
				`114`
				`230`

or for a collection of MCHs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM

`[ID=minid[-maxid]]`	`TRCMCH`	`[{ON|OFF}]`
		`[{LCL|GBL} [ICR {ON|OFF}]`
		`[ICRF {ON|OFF}]`
		`[OCR {ON|OFF}]`
		`[ICLR {ON|OFF}]`
		`[OCLR {ON|OFF}]]`
			`A`	`230`

`[IFN=minifn[-maxifn]]`				

or for all MCHs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM

	`TRCMCH`	`[{ALLON|ALLOFF}]`
		`[[GBL] [ICR {ON|OFF}]`
		`[ICRF {ON|OFF}]`
		`[OCR {ON|OFF}]`
		`[ICLR {ON|OFF}]`
		`[OCLR {ON|OFF}]]`		

This command is used to start (**ON**) or stop (**OFF**) *local* MCH event tracing, to enable *local* (**LCL**) or *global* (**GBL**) logging of MCH Incoming Call Requests (**ICR**), Outgoing Call Requests (**OCR**), Incoming Call Request Failures (**ICRF**), Incoming Clear Requests (**ICLR**) or Outgoing Clear Requests (**OCLR**) for the MCHs identified by the command modifiers (ID=, IFN=, RNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* MCH event tracing.

The ICR and OCR operands produce trace messages in SYSPRINT providing detailed information about the call request and PVC SETUP packets that HNAS receives (ICR) and sends (OCR). The output generated (see below for the message format) is affected by the BUILD statement **OPTIONS=ALRMSGTXT=SHORT** parameter. When SHORT is coded, inbound calls result in **NAS7730I** (**ICR**) or **NAS7731I** (**OCR**) messages which are sent to SYSPRINT

and to the operator's console. These messages are designed for use by programs that monitor the job log for call request information.

Note: All TRCMCH arguments can be entered at the same time with each one separated by space character. For example, TRCMCH ON LCL ICR OCR. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCMCH ON ON is rejected because ON is duplicated and TRCMCH ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting TRCMCH ON|OFF|...|ICLR|OCLR processing to the named MCH only.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), TRCMCH ON|OFF|...|ICLR|OCLR processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, TRCMCH ON|OFF|...|ICLR|OCLR processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, TRCMCH ON|OFF|INI|...|ICLR|OCLR processing is restricted to the selected PCEs. If **ID=0** is specified, TRCMCH ON|OFF|INI|...|ICLR|OCLR processing is for all PCEs. *The ID= modifier is only used if the RNM= modifier is not set. If ID= is also null, the command is rejected.*

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCMCH Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLON	ALLOFF	3
ALLOFF	ALLON	4
ICR ON	ICR OFF	5
ICR OFF	ICR ON	6
ICRF ON	ICRF OFF	7
ICRF OFF	ICRF ON	8
OCR ON	OCR OFF	9
OCR OFF	OCR ON	10
ICLR ON	ICLR OFF	11
ICLR OFF	ICLR ON	12
OCLR ON	OCLR OFF	13

PARAMETER	CONFLICT	ORDER
OCLR OFF	OCLR ON	14

- Enter **TRCMCH [ON]** to start *local* MCH event tracing for the MCHs identified by the command modifiers. This option provides LU and VC tracing for all sessions on the selected MCHs.

- Enter **TRCMCH OFF** to stop *local* MCH event tracing and *local* data logging (disables ICR, OCR, ICRF, ICLR and OCLR) for the MCHs identified by the command modifiers.

  Note: The **TRCMCH** command supports the ICRF, ICR, OCR, ICLR and OCLR arguments to allow tracing of Inbound Call failures (ICRF), all Inbound Calls (ICR), all Outbound Calls (OCR), all Inbound Clears (ICLR) and all Outbound Clears (OCLR). These operators can have *local* or *global* significance depending on whether the LCL or GBL argument is also specified.

- If **GBL** is specified, the ICRF, ICR, OCR, ICLR and/or OCLR trace flags are maintained in the HNAS common area (XFNASWA) and thus affect all MCHs. *Global* ICRF, ICR, OCR, ICLR and/or OCLR trace entries are only logged if all MCHs are also being traced (TRCMCH ALLON is in effect). **GBL is assumed if LCL is not set**.

  If you want ICRF, ICR, OCR, ICLR and/or OCLR to have *global* significance, we recommend that you should always enter GBL immediately after the TRCMCH command (e.g., TRCMCH GBL ICR ... OCLR) or set RNM= and ID= to null values.

- If **LCL** is specified, the ICRF, ICR, OCR, ICLR and/or OCLR trace flags are maintained in the MCH control blocks identified by the RNM=*mchname* or ID=*loid-hiid* command modifiers. *Local* ICRF, ICR, OCR, ICLR and/or OCLR trace entries are only logged if the MCH is also being traced (TRCMCH ON is in effect for the MCH). **RNM= or ID= must be set if LCL is specified**.

  If you want ICRF, ICR, OCR, ICLR and/or OCLR to have *local* significance, we recommend that you should enter LCL and omit GBL immediately after the TRCMCH command (e.g., TRCMCH LCL ICR ... OCLR) and set RNM= or ID= to a valid value.

- Enter **TRCMCH LCL ICR [ON]** to enable *local* Incoming Call Request logging for the MCHs identified by the command modifiers. Inbound PVC Setup packets will also be traced when **ICR** is entered.

- Enter **TRCMCH LCL ICR OFF** to disable *local* Incoming Call Request logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL ICRF [ON]** to enable *local* Incoming Call Request Failure logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL ICRF OFF** to disable *local* Incoming Call Request Failure logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL OCR [ON]** to enable *local* Outgoing Call Request logging for the MCHs identified by the command modifiers. Outbound PVC Setup packets will also be traced when **OCR** is entered.

- Enter **TRCMCH LCL OCR OFF** to disable *local* Outgoing Call Request logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL ICLR [ON]** to enable *local* Incoming Clear Request logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL ICLR OFF** to disable *local* Incoming Clear Request logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL OCLR [ON]** to enable *local* Outgoing Clear Request logging for the MCHs identified by the command modifiers.

- Enter **TRCMCH LCL OCLR OFF** to disable *local* Outgoing Clear Request logging for the MCHs identified by the command modifiers.

  Note: TRCMCH LCL ICR ... OCLR is treated the same as TRCMCH LCL ICR ... TRCMCH LCL OCLR.

- Enter **TRCMCH [GBL] ICR [ON]** to enable *global* Incoming Call Request logging for all MCHs. Inbound PVC Setup packets will also be traced when **ICR** is entered.

- Enter **TRCMCH [GBL] ICR OFF** to disable *global* Incoming Call Request logging for all MCHs.

- Enter **TRCMCH [GBL] ICRF [ON]** to enable *global* Incoming Call Request Failure logging for all MCHs.

- Enter **TRCMCH [GBL] ICRF OFF** to disable *global* Incoming Call Request Failure logging for all MCHs.

- Enter **TRCMCH [GBL] OCR [ON]** to enable g*lobal* Outgoing Call Request logging for all MCHs. Outbound PVC Setup packets will also be traced when **OCR** is entered.

- Enter **TRCMCH [GBL] OCR OFF** to disable g*lobal* Outgoing Call Request logging for all MCHs.

- Enter **TRCMCH [GBL] ICLR [ON]** to enable g*lobal* Incoming Clear Request logging for all MCHs.

- Enter **TRCMCH [GBL] ICLR OFF** to disable g*lobal* Incoming Clear Request logging for all MCHs.

- Enter **TRCMCH [GBL] OCLR [ON]** to enable g*lobal* Outgoing Clear Request logging for all MCHs.

- Enter **TRCMCH [GBL] OCLR OFF** to disable g*lobal* Outgoing Clear Request logging for all MCHs.

  Note: TRCMCH GBL ICR ... OCLR is treated the same as TRCMCH GBL ICR ... TRCMCH GBL OCLR.

- Enter **TRCMCH ALLON** to start *global* MCH event tracing. ALLON does not alter the data logging options (ICR, OCR, ICRF, ICLR and OCLR) currently in effect (equivalent to the TRCMCH [ALLON] start parameter).

- Enter **TRCMCH ALLOFF** to stop *global* MCH event tracing. ALLOFF does not alter the data logging options (ICR, OCR, ICRF, ICLR and OCLR) currently in effect (equivalent to the TRCMCH ALLOFF start parameter).

  Note: **TRCMCH** parameters **ICR** and **OCR** now produce expanded trace entries containing formatted inbound and outbound call request packet fields. The expanded information is located in the SYSPRINT output (not SYSCONS) and is only generated when the options are enabled or when **TRCLU** is enabled.

  The following is a sample of the information provided in the expanded trace entries, X.25 addresses are limited to 15 decimal digits. The first 32 bytes of facilities and call user data are displayed:

  For **TRCMCH ICR**:

  Call Request

```
NAS7718T 010.117.056.100(24321)  CALL REQ TO MCH MCH1
NAS7798T CALLED: 20360005        CALLING: 262199
NAS7798T FAC-LN: 06 FAC: 42070743 0707
NAS7798T USER DATA: 01000000
```

  When **OPTIONS=ALRMSGTXT=SHORT** is coded on build the above 4 messages are replaced by:

```
NAS7730I ED:called-dte-addr     ING:calling-dte-addr
CUD:4 bytes    FAC:ll(length)+6 bytes FAC data   MCH:mch-name
IP:ip-addr
```

  NAS7730I is sent to SYSPRINT and to the operator's console.

  PVC Setup

```
NAS7718T 010.117.056.100(24321)  PVCSETUP REQ TO MCH MCH1
NAS7798T PVCSTATUS=00 INIT LCN:NM 01:Serial0/1
         RESP LCN:NM 02:SerialMCH1
NAS7798T (SENDER) IN.WIN=4 OUT.WIN=4 IN.PSZ (2**N)=7
         OUT.PSZ (2**N)=7
```

  For **TRCMCH OCR**:

  Call Request

```
NAS7719T OUTBOUND CALL REQ GENERATED FOR LU MCH14001
         PLU=NASCTCP  REMOTE=R2CNOT1
NAS7798T CALLED: 10360003         CALLING: 1
NAS7798T FAC-LN: 06 FAC: 42070743 0202
NAS7798T USER DATA: 5C5C5C5C
```

When **OPTIONS=ALRMSGTXT=SHORT** is coded on build the above 4 messages are replaced by:

```
NAS7731I ED:called-dte-addr     ING:calling-dte-addr
CUD:4-bytes     FAC:length+6 bytes FAC data RMT:rmt-name
LU:lu-name
```

NAS7731I is sent to SYSPRINT and the operator's console.

PVC Setup

```
NAS7719T OUTBOUND PVCSETUP GENERATED FOR LU MCH14001
         PLU=NASCTCP  REMOTE=R2CNOT1
NAS7798T PVCSTATUS=00 INIT LCN:NM 01:SerialMCH1
         RESP LCN:NM 02:Serial0/1
NAS7798T (SENDER) IN.WIN=4 OUT.WIN=4 IN.PSZ (2**N)=7
         OUT.PSZ (2**N)=7
```

Note: The **ICR** trace generates records for all LLC types. The information is displayed is from the call request packet before any HNAS processing (STRIPRTEIN, etc.). The **OCR** trace generates records for LLC0, LLC4, and LLC5. The information displayed is from the call request packet when it is ready for link transfer. SYSPRINT records are generated whether or not TRCPRNT ON has been set. Records are not generated if the print limit (PRTLMT=) has been reached.

- To summarize, you may enter a TRCMCH parameter that requests a specific action to be performed as follows:

TRCMCH Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* MCH event tracing for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.**
OFF	**Deactivate *local* MCH event tracing and *local* data logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.**
LCL ICR ON or omitted	**Enable *local* Incoming Call Request logging and PVC Setup for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.**
LCL ICR OFF	**Disable *local* Incoming Call Request logging and PVC Setup for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.**

PARAMETER	ACTION
LCL ICRF ON or omitted	Enable *local* Incoming Call Request Failure logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL ICRF OFF	Disable *local* Incoming Call Request Failure logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL OCR ON or omitted	Enable *local* Outgoing Call Request and PVC Setup logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL OCR OFF	Disable *local* Outgoing Call Request and PVC Setup logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL ICLR ON or omitted	Enable *local* Incoming Clear Request logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL ICLR OFF	Disable *local* Incoming Clear Request logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL OCLR ON or omitted	Enable *local* Outgoing Clear Request logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
LCL OCLR OFF	Disable *local* Outgoing Clear Request logging for the MCHs identified by the ID= and IFN= modifiers or the RNM= modifier.
[GBL] ICR ON or omitted	Enable *global* Incoming Call Request logging and PVC Setup for all MCHs.
[GBL] ICR OFF	Disable g*lobal* Incoming Call Request logging and PVC Setup for all MCHs.
[GBL] ICRF ON or omitted	Enable *global* Incoming Call Request Failure logging for all MCHs.
[GBL] ICRF OFF	Disable g*lobal* Incoming Call Request Failure logging for all MCHs.
[GBL] OCR ON or omitted	Enable g*lobal* Outgoing Call Request and PVC Setup logging for all MCHs.
[GBL] OCR OFF	Disable g*lobal* Outgoing Call Request and PVC Setup logging for all MCHs.
[GBL] ICLR ON or omitted	Enable *global* Incoming Clear Request logging for all MCHs.
[GBL] ICLR OFF	Disable g*lobal* Incoming Clear Request logging for all MCHs.
[GBL] OCLR ON or omitted	Enable *global* Outgoing Clear Request logging for all MCHs.
[GBL] OCLR OFF	Disable g*lobal* Outgoing Clear Request logging for all MCHs.
ALLON	Activate *global* MCH event tracing and g*lobal* data logging (system wide).
ALLOFF	Deactivate *global* and all *local* MCH event tracing and data logging (system wide).

Example #1: To start *global* MCH event tracing for all MCHs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                    <- stop all tracing
TRCMCH ALLON                   <- start global MCH event tracing
```

Example #2: To start *local* MCH event tracing for a specific MCH when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                    <- stop all tracing
RNM=rmtname                    <- identify REMOTE
TRCMCH [ON]                    <- start local event tracing for MCH rmtname
TRCMCH LCL ICR                 <- log Incoming Call Requests for rmtname
TRCMCH LCL OCR                 <- log Outgoing Call Requests for rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS MCH event trace entries (ID=RTMC|RTCR|...) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCMCH:

1) HNAS activates with **TRCMCH ON** set in the absence of the **TRCMCH** start parameter.

2) The **TRCMCH [ON]** and **TRCMCH OFF** start parameters perform the same functions as the **TRCMCH ALLON** and **TRCMCH ALLOFF** console commands, respectively, for *global* MCH event tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCMCHX Command - MCH CTCP Trace Control (Privileged)

(changed for V2R2M0)

for a specific MCHX,

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[LUNM=*sluname*]	**TRCMCHX**	**[*sluname*]**	**C**	**230**
		[{ON|OFF}] **[DBK {ON|OFF}]**	**C**	**240**

or for the MCHXs on a specific MCH,

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*mchname*]	**TRCMCHX**	**[*mchname*]**	**C**	**230**
[LUN=*minlun* **[-*maxlun*]]**		**[{ON|OFF}]** **[DBK {ON|OFF}]**	**C**	**240**

or for a collection of MCHXs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCMCHX**	**[{ON|OFF}]** **[DBK {ON|OFF}]**	**C**	**240**
[IFN=*minifn* **[-*maxifn*]]**				
[LUN=*minlun* **[-*maxlun*]]**				

or for all MCHXs,

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCMCHX	**[{ALLON|ALLOFF}]**	**C**	**220**

This command is used to start (**ON**) or stop (**OFF**) *local* MCHX event tracing (GATE control session SLU activity) or to enable *local* data block logging (**DBK**) for the MCHXs identified by the command modifiers (ID=, IFN=, LUN=, RNM=, LUNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* MCHX event tracing (all GATE control session SLU activity).

Note: All TRCMCHX arguments can be entered at the same time with each one separated by space character. For example, TRCMCHX ON DBK ON. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCMCHX ON ON is rejected because ON is duplicated and TRCMCHX ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **LUNM=*sluname*** is specified, it overrides the **ID=**, **IFN=**, **LUN=** and **RNM=** values restricting TRCMCHX ON|OFF|DBK processing to the named MCHX (CTCP SLU) only.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting TRCMCHX ON|OFF|DBK processing to the named MCH only for the specified **LUN=** values. TRCMCHX ON|OFF|DBK processing will affect the MCHXs on the named MCH only.

- When **LUN=*minlun*[-*maxlun*]** is specified, TRCMCHX ON|OFF|DBK processing is restricted to the selected MCHXs on the selected MCHs. If **LUN=0** is specified, TRCMCHX ON|OFF|DBK processing is for all MCHXs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), TRCMCHX ON|OFF|DBK processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, TRCMCHX ON|OFF|DBK processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, TRCMCHX ON|OFF|DBK processing is restricted to the selected PCEs. If **ID=0** is specified, TRCMCHX ON|OFF|DBK processing is for all PCEs. *The ID= modifier is only used if the RNM= and LUNM= modifiers are not set. If ID= is also null, the command is rejected.*

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCMCHX Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLON	ALLOFF	3
ALLOFF	ALLON	4
DBK ON	DBK OFF	5
DBK OFF	DBK ON	6

- Enter **TRCMCHX [ON]** to start *local* MCHX event tracing for the MCHXs identified by the command modifiers. This option allows tracing of LU activity associated with selected GATE (LLC4) control sessions. There is an MCHX for each control session LU named by the LUNAME= operand.

- Enter **TRCMCHX OFF** to stop *local* MCHX event tracing and *local* data block logging (disables DBK) for the MCHXs identified by the command modifiers.

- Enter **TRCMCHX DBK [ON]** to enable *local* data block logging for the MCHXs identified by the command modifiers.
- Enter **TRCMCHX DBK OFF** to disable *local* data block logging for the MCHXs identified by the command modifiers.
- Enter **TRCMCHX ALLON** to start *global* MCHX event tracing and *local* data block logging (enables DBK) for all defined MCHXs (equivalent to the TRCMCHX start parameter).
- Enter **TRCMCHX ALLOFF** to stop *global* MCHX event tracing and *local* data block logging (disables DBK) for all defined MCHXs.
- To summarize, you may enter a TRCMCHX parameter that requests a specific action to be performed as follows:

TRCMCHX Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* MCHX event tracing for the MCHXs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers or the LUNM= modifier.**
OFF	**Deactivate *local* MCHX event tracing and *local* data block logging for the MCHXs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers or the LUNM= modifier.**
DBK ON or omitted	**Enable *local* data block logging for the MCHXs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers or the LUNM= modifier.**
DBK OFF	**Disable *local* data block logging for the MCHXs identified by the ID=, IFN= and LUN= modifiers or the RNM= and LUN= modifiers or the LUNM= modifier.**
ALLON	**Activate *global* MCHX event tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* MCHX event tracing (system wide).**

Example #1: To start *global* MCHX event tracing for all LUs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
TRCMCHX ALLON               <- start global MCHX event tracing
```

Example #2: To start *local* MCHX event tracing for a specific LU when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
LUNM=sluname                <- identify MCHX LU (CTCP SLU)
```

```
TRCMCHX [ON]              <- start local event tracing for MCHX LU sluname
```

Example #3: To start *local* MCHX event tracing for all the MCHXs associated with a TYPE=MCH|XTP REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP               <- stop all tracing
RNM=rmtname               <- identify REMOTE
LUN=0                     <- operate on all MCHXs
TRCMCHX [ON]              <- start local event tracing for MCHXs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS MCH event trace entries (ID=RTMC|RTCR|...) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCMCHX:

1) HNAS activates with **TRCMCHX ON** set in the absence of the **TRCMCHX** start parameter.

2) The **TRCMCHX [ON]** and **TRCMCHX OFF** start parameters perform the same functions as the **TRCMCHX ALLON** and **TRCMCHX ALLOFF** console commands, respectively, for *global* MCHX event tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCPCE Command - PCE Trace Control (Privileged)

(added for V2R3M0)
(changed for V2R4M0)

for a specific PCE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCPCE	**[{*pcestatic*|*pcename*}]**	**A**	**240**
		[{ON|OFF}]	**C**	**230**

or for a specific REMOTE

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*rmtname*]	**TRCPCE**	**[*rmtname*]**	**A**	**230**
		[{ON|OFF}]	**A**	**230**

or for a collection of PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCPCE**	**[{ON|OFF}]**	**A**	**230**

or for all PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCPCE	**[{ALLON|ALLOFF}]**	**A**	**230**

This command is used to start (**ON**) or stop (**OFF**) *local* PCE event tracing (includes I/O tracing, I/O buffer tracing, I/O data tracing and dispatcher tracing) for the PCEs identified by the command modifiers (ID= or RNM=) or to start (**ALLON**) or stop (**ALLOFF**) *global* PCE event tracing.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When the ***pcestatic*** (**LCLCONS**|**NASUTIL**|**PING**|**TIMER**|**XOTUTIL**|**XTPUTIL**) or ***pcename*** argument is provided, it directs TRCPCE processing to the identified PCE. For more information on *pcestatic*|*pcename* processing, please refer to section entitled 'PCE name as a Suffix Command Modifier' on page CONS-10.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** value currently in effect restricting TRCPCE ON|OFF processing to the named REMOTE only.

- When **ID=*minid*[-*maxid*]** is specified, TRCPCE ON|OFF processing is restricted to the selected PCEs. If **ID=0** is specified, TRCPCE ON|OFF processing is for all PCEs. The ID=

modifier is only used if the RNM= modifier is not set. If ID= is also null, the command is rejected.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCPCE Parameter Processing Matrix

PARAMETER	CONFLICT	COMPONENT AFFECTED	ORDER
ON	**OFF**	***pcestatic*|*pcename*|RNM=|ID=**	**1**
OFF	**ON**	***pcestatic*|*pcename*|RNM=|ID=**	**2**
ALLON	**ALLOFF**	**system wide**	**3**
ALLOFF	**ALLON**	**system wide**	**4**

- Enter **TRCPCE [ON]** to start *local* PCE event tracing for the PCEs identified by the command modifiers. *Local* PCE event tracing will be activated for all PCEs if **ID=0** and **RNM=** (null) is specified.

  Note: TRCPCE [ON] is treated as though TRCIO [ON], TRCBFR [ON], TRCDATA [ON] and TRCDISP [ON] were entered.

- Enter **TRCPCE OFF** to stop *local* PCE event tracing for the PCEs identified by the command modifiers. *Local* PCE event tracing will be deactivated for all PCEs if **ID=0** and **RNM=** (null) is specified.

  Note: TRCPCE OFF is treated as though TRCIO OFF, TRCBFR OFF, TRCDATA OFF and TRCDISP OFF were entered.

- Enter **TRCPCE ALLON** to start *global* PCE event tracing (equivalent to the TRCBFR, TRCDATA, TRCDISP and TRCIO start parameters).

  Note: TRCPCE ALLON is treated as though TRCIO ALLON, TRCIO ALLTYPES, TRCBFR ALLON, TRCBFR ALLTYPES, TRCDATA ALLON, TRCDATA ALLTYPES, TRCDISP ALLON and TRCDISP ALLTYPES were entered.

- Enter **TRCPCE ALLOFF** to stop *global* and all *local* PCE event tracing.

  Note: TRCPCE ALLOFF is treated as though TRCIO ALLOFF, TRCIO NOTYPES, TRCBFR ALLOFF, TRCBFR NOTYPES, TRCDATA ALLOFF, TRCDATA NOTYPES, TRCDISP ALLOFF and TRCDISP NOTYPES were entered.

- To summarize, you may enter a TRCPCE parameter that requests a specific action to be performed as follows:

TRCPCE Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* PCE tracing for the PCEs identified by the ID= or RNM= modifiers.**
OFF	**Deactivate *local* PCE tracing for the PCEs identified by the ID= or RNM= modifiers.**
ALLON	**Activate *global* PCE tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* PCE tracing (system wide).**

Example #1: To start *global* PCE event tracing for all PCEs when you are not sure of the the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
TRCPCE ALLON         <- start global event tracing
```

Example #2: To start *local* PCE event tracing for a specific PCE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
ID=pceid             <- identify PCE
TRCPCE [ON]          <- start local event tracing for PCE pceid
```

Example #3: To start *local* PCE event tracing for all the PCEs associated with a TYPE=XOT REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
RNM=rmtname          <- identify REMOTE
TRCPCE [ON]          <- start local event tracing for PCEs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS PCE trace entries (ID=1|2|3|45|6|7|A|B) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCPCE:

1) The TRCPCE ALLON|ALLOFF commands alter the PCE types that currently in effect for *global* PCE I/O, I/O buffer, I/O data and dispatcher tracing.

2) HNAS activates with **TRCPCE OFF** set. There is no equivalent start parameter.

TRCPRNT Command - SYSPRINT Trace Control (Privileged)

(changed for V2R2M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCPRNT	**[{ON|OFF}]**		

This command is used to start (**ON**) or stop (**OFF**) SYSPRINT trace logging. Trace entries are always logged in the HNAS memory resident trace table.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCPRNT Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**

- Enter **TRCPRNT [ON]** to force trace entries that are logged in the HNAS memory resident trace table to also be written to the SYSPRINT log (if **PRNT ON** is in effect).

- Enter **TRCPRNT OFF** to inhibit writing trace entries to the SYSPRINT log.

Caution: If the number of records written to the SYSPRINT data set reaches the limit specified by the PRTLMT operand on the BUILD definition statement, HNAS will stop logging output. The SYSPRINT data set can be the JES spool or a z/OS, OS/390 or MVS sequential data set. A message will be written to the local console each time 4096 records are lost.

General notes for TRCPRNT:

1) HNAS activates with **TRCPRNT OFF** set in the absence of the **TRCPRNT** start parameter.

2) The **TRCPRNT [ON]** and **TRCPRNT OFF** start parameters perform the same function as the **TRCPRNT [ON]** and **TRCPRNT OFF** console commands, respectively, for *global* trace print control. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

3) HNAS activates with a number default traces enabled so that potential configuration problems can be captured and reviewed should a problem occur during initial product testing. Please refer to the **TRCLU [ON]**, **TRCLU MINDATA**, **TRCVC [ON]**, **TRCVC MINDATA**, **TRCMCH [ON]** and **TRCMCHX [ON]** console commands which represent the defaults that are set during processing of the **PARM=** start parameter list. If you plan to enable **TRCPRNT ON**, please be aware that added overhead will result from the trace activity. Once your environment has completed initial testing we suggest that you disable the

default traces by specifying the following values in the **PARM=** operand: **TRCLU OFF**, **TRCVC OFF**, **TRCMCH OFF** and **TRCMCHX OFF**.

TRCSUBR Command - Subroutine Call Trace Control (Privileged)

(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCSUBR	**[{ON|OFF}]** **[{ALLEVENTS|NOEVENTS|** **CONS|NETV|TCP|TMR|** **UTIL|XOT|XTP}]**	**A**	**240**

This command is used to start (**ON**) or stop (**OFF**) subroutine call tracing and/or to identify the type(s) of events (**ALLEVENTS|CONS|MCH|NETV|PCE|TCP|VTAM**) for which subroutine call tracing is applicable.

Note: All TRCSUBR arguments can be entered at the same time with each one separated by space character. For example, TRCSUBR ON TCP. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCSUBR ON ON is rejected because ON is duplicated and TRCSUBR ON OFF is rejected because ON and OFF conflict.

- The event list parameters (**ALLEVENTS|CONS|MCH|NETV|PCE|TCP|VTAM**) allow you to control subroutine call tracing for **all** events that HNAS processes. They also allow you to identify specific events for which subroutine calls are to be traced. The **TRCSUBR** start parameter and console command allow an event list to be provided in addition to the normal ON|OFF arguments. Event list values are provided so that subroutine call traces can be filtered by the event currently being processed. This reduces the number of unwanted trace entries being logged.

  When TRCSUBR is in effect, every subroutine within HNAS logs a number of trace entries. Some are very useful but others are not really required. What is necessary to eliminate unwanted TRCSUBR entries is the ability to filter subroutine calls based on the event(s) being processed. For example, the TRCPCE command is used to log TCP/IP related events. To coordinate TCP/IP subroutine calls with these events requires filtering TRCSUBR traces for TCP/IP related calls only. Currently, HNAS waits on the following 6 events:

  TCP - TCP/IP interrupt completions
  VTAM - VTAM interrupt completions
  MCH - REMOTE TYPE=MCH service
  NETV - NETVIEW interrupt completions
  CONS - CONSOLE interrupt completions
  PCE - Miscellaneous task service

  The TRCSUBR start parameter and console command will accept one or more of these events to be specified so that subroutine call traces are logged only when the selected event(s) are being processed. This means that only subroutine calls associated with the selected event(s) will generate trace entries.

  Note: TRCSUBR ***eventlist*** logic was introduced into 240 via APAR 2400108.

- Subroutine call tracing is handy when used in conjunction with PCE dispatcher and I/O tracing because you can observe the sequence of events that occur before I/O operations are started, during their execution and after their completion.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCSUBR Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLEVENTS	NOEVENTS	3
NOEVENTS	ALLEVENTS	4
CONS	none	5
MCH	none	6
NETV	none	7
PCE	none	8
TCP	none	9
VTAM	none	10

- Enter **TRCSUBR [ON]** to start subroutine call tracing.

- Enter **TRCSUBR OFF** to stop subroutine call tracing.

- Enter **TRCSUBR ALLEVENTS** to set all events for subroutine call tracing.

- Enter **TRCSUBR NOEVENTS** to reset all events for subroutine call tracing.

- Enter **TRCSUBR {CONS|MCH|NETV|PCE|TCP|VTAM}** to set one or more specific events for subroutine call tracing.

  Note: To restrict subroutine call tracing to a specific event or events, for example TCP and VTAM, enter **TRCSUBR NOEVENTS TRCSUBR TCP VTAM**.

- To summarize, you may enter a TRCSUBR parameter that requests a specific action to be performed as follows:

TRCSUBR Action Matrix

PARAMETER	ACTION
ON or omitted	Activate subroutine call tracing.

PARAMETER	ACTION
OFF	**Deactivate subroutine call tracing.**
ALLEVENTS	**Allow subroutine call tracing for all events.**
NOEVENTS	**Inhibit subroutine call tracing for all events.**
CONS	**Allow subroutine call tracing for CONSOLE events.**
MCH	**Allow subroutine call tracing for REMOTE TYPE=MCH related events.**
NETV	**Allow subroutine call tracing for NETVIEW events (future support).**
PCE	**Allow subroutine call tracing for PCE (process control) related events.**
TCP	**Allow subroutine call tracing for TCP/IP events.**
VTAM	**Allow subroutine call tracing for VTAM events.**

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS subroutine call trace entries (ID=8) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCSUBR:

1) HNAS activates with **TRCSUBR OFF** set in the absence of the **TRCSUBR** start parameter.

2) The **TRCSUBR [ON]** and **TRCSUBR OFF** start parameters perform the same function as the **TRCSUBR [ON]** and **TRCSUBR OFF** console commands, respectively, for *global* subroutine call tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

3) ON is assumed if TRCSUBR is entered with no argument.

4) If **NOEVENTS** is in effect when **ON** is set, **ALLEVENTS** is forced.

TRCTASK Command - Task WAIT/POST Trace Control (Privileged)

(new for V2R1M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCTASK	**[{ON|OFF}]**	**N**	**210**

This command is used to start (**ON**) or stop (**OFF**) task WAIT/POST tracing.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCTASK Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**

- Task WAIT/POST tracing is handy when used in conjunction with other HNAS tracing because you can observe the frequency of HNAS main task dispatching.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of HNAS task WAIT trace entries (ID=E) and POST trace entries (ID=F) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCTASK:

1) HNAS activates with **TRCTASK OFF** set in the absence of the **TRCTASK** start parameter.

2) The **TRCTASK [ON]** and **TRCTASK OFF** start parameters perform the same function as the **TRCTASK [ON]** and **TRCTASK OFF** console commands, respectively, for *global* task tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCTRAP Command - Trace Trap Suspension Control (Privileged)

(added for V2R1M0)
(changed for V2R3M0)
(changed for V2R4M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCTRAP	[{{SUSP|SUSPEND}| {RSME|RESUME}}]	A A	240 230
		[{SNAP|RSMESNAP}]		
		[RSMEEXEC]	N	240
		[RSMEALL]		
		[SHOW]		
		[ALRMLIST={?|*| (*id*1/*dat*1/*sod*1, ..., *id*n/*dat*n/*sod*n)}]	C	240
		[RCVLIST={?|*| (*prot*1||*pkt*1, ..., *prot*n||*pkt*n)}]		
		[XMTLIST={?|*| (*prot*1||*pkt*1, ..., *prot*n||*pkt*n)}]		
		[TRAPACTION={?|*| ALL|NONE| ({SUSP|NOSUSP}, {SNAP|NOSNAP}, EXEC=*ddname*)]	N	240

This command is used to suspend active tracing (**SUSP|SUSPEND**), resume suspended tracing (**RSME|RESUME**), take a snapshot dump (**SNAP**), allow another snapshot dump after a trace trap 'hit' (**RSMESNAP**), allow another command list execution after a trace trap 'hit' (**RSMEEXEC**), resume suspended tracing, allow another snapshot dump and allow another command list execution after a trace trap hit (**RSMEALL = RSME** + **RSMESNAP** + **RSMEEXEC**), display all trace trap parameters that are currently active (**SHOW**), set or display specific trace trap parameters that will be in effect (**ALRMLIST=**, **RCVLIST=** or **XMTLIST=**) and/or set the action to be taken on a trace trap hit (**TRAPACTION=**).

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCTRAP Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ALRMLIST=*	**none**	**1**
ALRMLIST=*list*	**none**	**2**

PARAMETER	CONFLICT	ORDER
RCVLIST=*	**none**	**3**
RCVLIST=*list*	**none**	**4**
XMTLIST=*	**none**	**5**
XMTLIST=*list*	**none**	**6**
TRAPACTION=*	**none**	**7**
TRAPACTION=*list*	**none**	**8**
ALRMLIST=?	**none**	**9**
RCVLIST=?	**none**	**10**
XMTLIST=?	**none**	**11**
SUSP|SUSPEND	**RSME|RESUME**	**12**
RSME|RESUME	**SUSP|SUSPEND**	**13**
SNAP	**RSMESNAP|RSMEALL**	**14**
RSMESNAP|RSMEALL	**SNAP**	**15**
RSMEEXEC	**none**	**16**
RSMEALL	**SNAP**	**17**

Trap Control Commands

- Enter **TRCTRAP SUSP** to manually suspend active tracing.
- Enter **TRCTRAP RSME** to resume tracing that has been suspended.
- Enter **TRCTRAP SNAP** to manually take a snapshot dump which will be logged in SYSPRINT.
- Enter **TRCTRAP RSMESNAP** to allow another snapshot dump is to be taken after one was triggered as the result of a trace trap hit.
- Enter **TRCTRAP RSMEEXEC** to allow another command list to be executed after one was triggered as the result of a trace trap hit.
- Enter **TRCTRAP RSMEALL** to resume suspended tracing, allow another snapshot dump to be taken and allow another command list to be executed after they were triggered as the result of a trace trap hit.

  Note: RSMESNAP|RSMEALL is not required after a manual SNAP dump but is required when a SNAP dump occurs as the result of a trap hit.
- Enter **TRCTRAP SHOW** to display all trace trap parameters currently in effect (treated the same as ALRMLIST=?, RCVLIST=?, XMTLIST=? and TRAPACTION=?).

Note: In addition to the list display produced by TRCTRAP ALRMLIST=?, TRCTRAP RCVLIST=?, TRCTRAP XMTLIST=?, TRCTAP TRAPACTION=? or TRCTRAP SHOW, the following trace state information is displayed:

```
TRACE STATE: IDLE                  <- logging inactive
             LOGGING               <- logging active
             SUSPENDED BY COMMAND  <- logging suspended
             SUSPENDED BY TRAP     <- logging suspended
```

Trap Filtering Overview

The ALRMLIST=, RCVLIST= and/or XMTLIST= command parameters are used, respectively, to specify a list of alarm message identifiers and optional embedded data, a list of partial or complete receive packet data and/or a list of partial or complete transmit packet data that will be used as trapping parameters in order to automatically suspend HNAS internal tracing when any list entry is detected.

In many cases, the only way to resolve a problem is to run HNAS traces. However, sometimes it is difficult to capture the cause of a problem in the HNAS internal trace table when tracing has to be stopped manually. If a problem is intermittent, TRCPRNT (SYSPRINT trace logging) is not a viable option because it can create an inordinate amount of SYSPRINT and consume valuable CPU cycles. Internal tracing is the next best option but it must be stopped as soon as possible after the problem has occurred. This can be difficult and, in some cases, may even be impossible.

To circumvent delays inherent with manual intervention, HNAS has been modified to accept the TRCTRAP= configuration operand and TRCTRAP console command. These facilities provide the ability to suspend internal tracing (and TRCPRNT) based on an alarm message ID match (refined by optional embedded data matching), an input packet match or an output packet match. Multiple alarm IDs and packets may be specified so that a collection of values may be monitored (trapped).

Once internal trace and TRCPRNT activity is suspended (the NAS0050A message generated), HNAS must be shutdown with **QY*/password*** so that the internal trace table can be formatted. When SUSP is specified as one of the TRAPACTIONs, the trace is suspended (frozen) when the trap is triggered so HNAS need not be shutdown immediately but can be shutdown at your convenience.

If the trap was performed with TRCPRNT enabled and SYSPRINT is the JES spool (SYSOUT=* for example), you can use SDSF to save the SYSPRINT to a dataset so that it can sent to us for analysis. When saving SYSPRINT to a dataset, DCB parameters of RECFM=FBA, LRECEL=133 and BLKSIZE=3990 should be used.

If the trap was performed with TRCPRNT enabled and SYSPRINT is already a dataset, the SYSPRINT file can be manually switched so that the current captured trace can be sent to us for analysis (see **PRNT CLOSE|OPEN *ddname* or PRNT CLSOPN *ddname*** console command for additional information on HNAS SYSPRINT dataset switching). This will allow you to send us the closed SYSPRINT file without having to shutdown HNAS. If multiple

SYSPRINT datasets are used, make sure that DISP=SHR is specified for all so that they can be accessed by ISPF or another program (like IEBCOPY or IEBGENER). As with the JES spool, SYSPRINT datasets should use DCB parameters of RECFM=FBA, LRECEL=133 and BLKSIZE=3990.

Trap Alarm Filtering Commands

Enter TRCTRAP ALRMLIST=(*id*1/*dat*1/*sod*1,...,*id*n/*dat*n/*sod*n) to set trap alarm filter IDs with optional embedded data.

ALRMLIST=(*list*) specifies a list of trap alarm filer IDs that when detected will cause the TRAPACTIONs to be automatically executed. If optional embedded data is supplied, an alarm ID match will not invoke the TRAPACTIONs unless there is also a match on the supplied data.

You may specify from 1 to 16 ***id*i** values (e.g, NAS1001I). Wildcard characters (*) are also accepted. Any character in a wildcard position is considered a matched character.

If embedded data (***dat*i**) is supplied, ***it must be enclosed in quotes if it contains spaces or foward slashes (/)***. The specified data, which can be from 1 to 70 characters in length, is used to further restrict the message search to specific data content. For example, if *id*i=NAS7707W and *dat*i=STATUS=0C are given, the TRAPACTIONs will not be invoked unless the alarm ID is NAS7707W and the message contains the text string STATUS=0C. This message is issued when PVC Setup fails. For these *id*i and *dat*i values, The ALRMLIST syntax would be ALRMLIST=(NAS7707W/STATUS=0C).

Note: If *dat*i does not contain spaces or forward slashes, it does not have to be specified within quotes, but can be if you wish. ABCDEF and 'ABCDEF' are treated the same.

Note: *dat*i may not include embedded quotes. For example, ABC'DEF or ABC''DEF or 'ABC'DEF' or 'ABC''DEF' are not permitted.

To reduce the search time for embedded data, you can also specify a start of data offset (***sod*i**) which can range from 0 to 128. The start of data offset allows the embedded data search to start at a specific offset within the message which reduces CPU utilization when looking for a match. The *sod*i value is relative to the first character of the message, that is, the beginning of the message ID (NASnnnns). If no *sod*i value is supplied, the *dat*i search starts at the beginning of the message. If the *sod*i value plus the *dat*i length exceeds the message length (which implies that *dat*i cannot be in the message), a trap hit is assumed for the message ID only. As an example, the NAS7707W message has the following form:

```
          1         2         3         4         5         6         7         8
012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
NAS7707W XOT PVC SETUP FROM 010.117.056.100(29880) CAN'T START SESSION.  STATUS=0C
```

To reduce the STATUS=0C search time within the NAS7707W message, a *sod*i offset value of 73 would be used: ALRMLIST=(NAS7707W/STATUS=0C/73).

Note: Logic to accept *dat*i as a quoted string ('*dat*i') was introcuced into 240 by APAR 2400108. Prior to this APAR, quoted strings were not permitted for *dat*i. If *dat*i contained a

forward slash (/), the slash would be treated as a delimiter making the data that follows it being treated as the *sod*i suboperand. This would cause the entire ALRMLIST value to be ignored and an error condition to be set. For example, when the following TRCTRAP console command was entered:

```
TRCTRAP ALRMLIST=(NAS3799I/DIAG=000/195)
```

the following error message was issued and the request was rejected:

```
NASC532E PARAMETER DATA INVALID: IAG=000/195)...,
         TRCTRAP COMMAND ABORTED
```

This occurred because the *dat*i value DIAG=000/195 contained a forward slash making the 195 that follows the slash treated as a *sod*i value. Since 195 is too large for a *sod*i value, the command was rejected.

Effective with APAR 2400108, the BUILD TRCTRAP operand processor and the TRCTRAP console command processor will now accept the *dat*i suboperand of the ALRMLIST operand as a quoted string so that *all* data within the string is treated as data. This includes spaces, forward slashes and so on. For the ALRMLIST operand described above, the following is now allowed:

```
TRCTRAP ALRMLIST=(NAS3799I/'DIAG=000/195')
```

In this case, the forward slash in DIAG=000/195 is no longer treated as an suboperand separator but as part of the message data.

- Enter TRCTRAP ALRMLIST=* to force the trap alarm filter array to be cleared. The array count is set to zero and all trap alarm filter IDs are set to null values.

- Enter TRCTRAP ALRMLIST=? to display the trap alarm filter IDs that are currently in effect. The following is example of the information produced by the `TRCTRAP ALRMLIST=?` command.

```
ALRMLIST=NAS3****
         NAS*****
```

Note: The ALRMLIST= parameter of the TRCTRAP command should not be confused with the FILTER= parameter of the ALARM command. While they both list alarm IDs in a similar fashion, their functions are completely independent.

Trap Packet Filtering Commands

- Enter **TRCTRAP RCVLIST=(*prot*1||*pkt*1,...,*prot*n||*pkt*n)** or
  TRCTRAP XMTLIST=(*prot*1||*pkt*1,...,*prot*n||*pkt*n), as appropriate, to

supply a list of trap packet data filters that when detected will cause the TRAPACTIONs to be automatically executed.

You may specify ***proti||pkti*** values in the RCVLIST= and XMTLISTS= parameters. The protocol value (***proti***) must be 00 for XTP or 04 for XOT. The packet data (***pkti***) can be from 1 to n hex bytes in length. The framing characters X'' must be omitted. All ones (FF) are accepted as wildcards. Any value in a wildcard byte position is considered a match. Data is compared only for the length of the packet (n) you specify starting with the packet type byte.

A maximum of 512 bytes of protocol and packet data may be specified for the RCVLIST= and XMTLIST= suboperands which include a length byte (n) for each packet.

Note: PAD sessions operating with non-space parity will encounter data mismatches with the filter list when non-qualified data is specified. You may specify the same data with different parity settings as required.

- Enter **TRCTRAP RCVLIST=*** or **TRCTRAP XMTLIST=***, as appropriate, to force the corresponding trap packet filter array to be cleared. The corresponding array count is set to zero and all trap packet filters are set to zero values.

- Enter **TRCTRAP RCVLIST=?** or **TRCTRAP XMTLIST=?**, as appropriate, to display the trap packet filters that are currently in effect. The following is example of the information produced by the `TRCTRAP RCVLIST=?` command.

```
RCVLIST=041B0502
        041309A5
```

Note: A match on any value in the ALRMLIST=, RCVLIST= or XMTLIST= operands will cause the TRAPACTIONs to be automatically executed. The first match in any of these operands is the winner. Multiple operands with multiple values are provided to allow for the greatest flexibility.

Trap Filtering Action Commands

- Enter **TRCTRAP TRAPACTION=(ALL|NONE|{SUSP|NOSUSP},{SNAP|NOSNAP},EXEC=*ddname*)** to specify the action to be taken on a trap hit. The TRAPACTION= subparameter is very handy when you are unable to capture the cause of certain problems because systems operation is automated or unattended. Many times it is difficult and sometimes impossible to trap an error and SHUTDOWN HNAS before debugging control block information is lost. This may be the case even when operations personnel are present and is certainly the case when no one is around. The TRAPACTION= suboperand provides functionality that eliminates the need for an operator initiated SHUTDOWN or a forced ABEND in order to collect additional diagnostic information.

- **TRAPACTION=SUSP**, which is the default trace trap action, will cause HNAS tracing and SYSPRINT trace logging to be suspended when a trace trap hit occurs.

- **TRAPACTION=SNAP** will cause a snapshot of all HNAS control blocks and trace entries to be logged in SYSPRINT (SNAP dump) when a trace trap hit occurs.

- **TRAPACTION=EXEC=*ddname*** will cause the command list identified by the specified *ddname* to be executed when a trace trap hit occurs. The following is a sample command list that can be provided via **TRAPACTION=EXEC=*ddname***:

```
* THIS IS A SAMPLE CONSOLE COMMAND LIST FILE. COMMENTS CAN
* BE SUPPLIED THAT START WITH '*' OR ';' IN RECORD COLUMN 1.
* COMMENTS ON INDIVIDUAL LINES CAN BE SUPPLIED THAT START
* WITH ';'.  CONSOLE COMMANDS WILL BE EXTRACTED FROM EACH
* RECORD FROM THE FIRST NON-BLANK CHARACTER TO THE LAST
* NON-BLANK CHARACTER ON EACH LINE.  THE SEARCH FOR THE
* FIRST NON-BLANK CHARACTER IS EXECUTED FOR NON-COMMENT
* RECORDS ONLY (*|; IS NOT IN CC1).  THE SEARCH FOR THE
* LAST NON-BLANK CHARACTER IS AFFECTED AFTER THE LINE
* COMMENT STARTING DELIMITER (;) IS FOUND.
* |<-- FIRST NON-BLANK
* |
* |           |<-------- LAST NON-BLANK
* |           |
* V           V
  TRCTRAP SUSP      ; 1ST QUEUED COMMAND TO SUSPEND TRACING
  TRCTRAP SNAP      ; 2ND QUEUED COMMAND TO TAKE A SNAP DUMP
  TRCTRAP RSMEALL   ; 3RD QUEUED COMMAND TO RESUME TRACE TRAPPING
```

  The SYSPRINT log now contains the trapped event and a new trap has been started. For more information on command list processing, see Input Rules, Command Lists on page CONS-9 and the EXEC command description or page CONS-139.

- **TRAPACTION=NONE** prevents any action from being taken when a trace trap hit occurs. This effectively disables the trap function until a real trap action is provided. TRAPACTION=NONE is the same as TRAPACTION=(NOSUSP,NOSNAP,EXEC=).

- **TRAPACTION=ALL** will cause tracing and SYSPRINT logging to be suspended and a SNAP dump to be taken when a trace trap hit occurs. TRAPACTION=ALL is the same as TRAPACTION=(SUSP,SNAP). TRAPACTION=ALL does not also include EXEC because a *ddname* argument is required for EXEC.

  Note: If multiple actions are required when a trap hit occurs, all must be specified in the TRAPACTION= operand. For example, if tracing and SYSPRINT logging are to be suspended and a SNAP dump is to be taken on a trap hit, specify TRCTRAP TRAPACTION=ALL or TRCTRAP TRAPACTION=(SUSP,SNAP). If a single action is to be performed when a trap hit occurs, specify only that action for the TRAPACTION= operand.

  Note: If a trace trap hit occurs and the specified action(s) are taken, a subsequent trace trap hit will not repeat the action(s) until the associated resume function is performed, that is, until the TRCTRAP RSME, TRCTRAP RSMESNAP, TRCTRAP RSMEEXEC or TRCTRAP RSMEALL command is entered.

  Note: When multiple trap actions are specified, they are processed in the following order:

  1) SUSP
  2) SNAP

3) EXEC *ddname*

- Enter **TRCTRAP TRAPACTION=*** to set the trace trap actions to be set to their off state. Specifying TRAPACTION=* is the same as specifying TRAPACTION=(NOSUSP,NOSNAP,EXEC=) or TRAPACTION=NONE.

- Enter **TRCTRAP TRAPACTION=?** to display the trap actions that are currently in effect. The following is example of the information produced by the `TRCTRAP TRAPACTION=?` command.

```
TRAPACTION=SUSP
           NOSNAP
           EXEC=CMDLIST1
```

Example #1: Trap multiple events:

```
 TRCTRAP ALRMLIST=(NAS3701W,  <- LU ACB open failed issued
                   NAS3798I)  <- LU starting session issued
         RCVLIST=(041B0502,   <- Reset 05/02 received
                  041309A5,   <- Clear 09/A5 received
                  04F1FFFF)   <- any Diag packet received
         XMTLIST=(041BFFFF,   <- any Reset packet sent
                  0413FFFF)   <- any Clear packet sent
```

Example #2: Trap PAD logon request:

```
TRCTRAP RCVLIST=(04FF6CA063F0,  <- l cp (LC even)
                 04FFEC20E370,  <- l cp (LC odd)
                 04FF6C206370,  <- l cp (LC none)
                 04FFCCA0C350,  <- L CP (UC even)
                 04FF4C2043D0,  <- L CP (UC odd)
                 04FF4C204350)  <- L CP (UC none)
```

Example #3: Trap QLLC Terminate-Self PIU:

```
TRCTRAP RCVLIST=(04FF2C0000FFFFFF0B8000010683) <- TERM-SELF
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

General notes for TRCTRAP:

1) The current tracing parameters are not altered when tracing is suspended. To restart trace logging, you need to issue the **TRCTRAP RSME** or **TRCALL RSME** console command.

2) When tracing is suspended, **TRCPRNT** trace entry logging in SYSPRINT is also suspended.

3) Since **TRCTRAP** processing is global (not directly associated with enabled trace options), it is important to have the appropriate traces activated so that the events leading up to the suspended trace will be captured in the internal trace table. For example, if a RCVLIST= or XMTLIST= is provided, TRCVC MAXDATA or TRCVC MINDATA should be active.

4) When a **TRCTRAP** hit occurs, a NAS0050A messages is issued and the specified TRAPACTION is taken.

5) **TRCTRAP** logic was added primarily as a tool for Comm-Pro in order to provide a way of simplifying problem diagnosis. In the past, Comm-Pro had to develop custom modifications to trap certain events. The TRCTRAP logic now standardizes this processing. In most cases, Comm-Pro will be advising customers on what to enter for TRCTRAP parameters rather than giving them a custom trap modification. This description is intended to provide some information about the TRCTRAP function. We do not, however, expect customers to use TRCTRAP functions without Comm-Pro involvement.

TRCVC Command - VC Trace Control (Privileged)

(changed for V2R2M0)

for the VCs on a specific MCH,

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*mchname*]	**TRCVC**	**[*mchname*]**	**C**	**230**
[VCN=*minvcn*[-*maxvcn*]]		**[{ON|OFF]** **[DBK {ON|OFF}** **[{MAXDATA|MINDATA|NODATA}]**	**C** **C** **A**	**220** **240** **220**

or for a collection of VCs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCVC**	**[{ON|OFF}]** **[DBK {ON|OFF}]** **[{MAXDATA|MINDATA|NODATA}]**	**C** **C** **A**	**220** **240** **220**
[IFN=*minifn*[-*maxifn*]]				
[VCN=*minvcn*[-*maxvcn*]]				

or for all VCs,

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCVC	**[{ALLON|ALLOFF}]** **[{MAXDATA|MINDATA|NODATA}]**	**C** **A**	**220** **220**

This command is used to start (**ON**) or stop (**OFF**) *local* VC event tracing or to enable *local* data block logging (**DBK**) for the VCs identified by the command modifiers (ID=, IFN=, VCN=, RNM=), to start (**ALLON**) or stop (**ALLOFF**) *global* VC event tracing or to control the amount of data (**MAXDATA**|**MINDATA**|**NODATA**) that is logged for the VCs currently being traced.

Note: All TRCVC arguments can be entered at the same time with each one separated by space character. For example, TRCVC ON MAXDATA. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCVC ON ON is rejected because ON is duplicated and TRCVC ON OFF is rejected because ON and OFF conflict.

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting TRCVC ON|OFF|DBK processing to the named MCH only for the specified **VCN=** values. TRCVC ON|OFF|DBK processing will affect the VCs on the named MCH only.

- When **VCN=*minvcn*[-*maxvcn*]** is specified, TRCVC ON|OFF|DBK processing is restricted to the selected VCs on the selected MCHs. If **VCN=0** is specified, TRCVC ON|OFF|DBK pro-

cessing is for all VCs on the selected MCHs.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), TRCVC ON|OFF|DBK processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, TRCVC ON|OFF|DBK processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[-*maxid*]** is specified, TRCVC ON|OFF|DBK processing is restricted to the selected PCEs. If **ID=0** is specified, TRCVC ON|OFF|DBK processing is for all PCEs. *The ID= modifier is only used if the RNM= modifier is not set. If ID= is also null, the command is rejected.*

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCVC Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLON	ALLOFF	3
ALLOFF	ALLON	4
DBK ON	DBK OFF	5
DBK OFF	DBK ON	6
MAXDATA	MINDATA|NODATA	7
MINDATA	MAXDATA|NODATA	8
NODATA	MINDATA|MAXDATA	9

- Enter **TRCVC [ON]** to start *local* VC event tracing for the VCs identified by the command modifiers.

- Enter **TRCVC OFF** to stop *local* VC event tracing and *local* data block logging (disables DBK) for the VCs identified by the command modifiers.

- Enter **TRCVC DBK [ON]** to enable *local* data block logging for the VCs identified by the command modifiers. This is equivalent to MAXDATA on a VC basis rather than a system wide basis.

- Enter **TRCVC DBK OFF** to disable *local* data block logging for the VCs identified by the command modifiers. This is equivalent to MAXDATA on a VC basis rather than a system wide basis.

- Enter **TRCVC ALLON** to start *global* VC event tracing (equivalent to the TRCVC start parameter).

- Enter **TRCVC ALLOFF** to stop *global* and *local* VC event tracing and *local* data block logging (disables DBK) for all defined VCs.

- Enter **TRCVC MAXDATA** to enable *global* data block logging with a maximum amount of data being logged for the VCs currently being traced.

- Enter **TRCVC MINDATA** to enable *global* data block logging with a minimum amount of data being logged for the VCs currently being traced.

- Enter **TRCVC NODATA** to disable *global* data block logging for the VCs currently being traced. This prevents any additional data being logged when a VC event trace entry is recorded.

  Note: For HNAS release V2R2M0, the MAXDATA, MINDATA and NODATA parameters also started event tracing for all VCs defined to HNAS. This precluded their use to control data logging for a single VC. For this reason, the VC event start function has been removed from these parameters. Global VC event tracing must now be started and stopped using the ALLON and ALLOFF parameters, respectively.

- To summarize, you may enter a TRCVC parameter that requests a specific action to be performed as follows:

  TRCVC Action Matrix

PARAMETER	ACTION
ON or omitted	**Activate *local* VC event tracing for the VCs identified by the ID=, IFN= and VCN= modifiers or the RNM= and VCN= modifiers.**
OFF	**Deactivate *local* VC event tracing and *local* data block logging for the VCs identified by the ID=, IFN= and VCN= modifiers or the RNM= and VCN= modifiers.**
DBK or omitted	**Enable *local* data block logging for the VCs identified by the ID=, IFN= and VCN= modifiers or the RNM= and VCN= modifiers.**
DBK OFF	**Disable *local* data block logging for the VCs identified by the ID=, IFN= and VCN= modifiers or the RNM= and VCN= modifiers.**
ALLON	**Activate *global* VC event tracing (system wide).**
ALLOFF	**Deactivate *global* and all *local* VC event tracing (system wide).**
MAXDATA	**Activate *global* data block logging with a maximum amount of data being logged for the VCs currently being traced.**
MINDATA	**Activate *global* data block logging with a minimum amount of data being logged for the VCs currently being traced.**
NODATA	**Deactivate *global* data block logging for the VCs currently being traced.**

Example #1: To start *global* VC event tracing for all VCs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
TRCVC ALLON                 <- start global VC event tracing
```

Example #2: To start *local* VC event tracing for a specific VC when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
VCN=vcn                     <- identify VC
TRCVC [ON]                  <- start local event tracing for VC vcn
```

Example #3: To start *local* VC event tracing for all the VCs associated with a TYPE=MCH|XTP REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP                 <- stop all tracing
RNM=rmtname                 <- identify REMOTE
VCN=0                       <- operate on all VCs
TRCVC [ON]                  <- start local event tracing for VCs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS VC event trace entries (ID=LSRS|LSCL|...) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCVC:

1) Entering **TRCVC OFF** will terminate *local* event tracing and *local* data block (**DBK**) logging.

2) Entering **TRCVC ON|OFF|DBK|ALLON|ALLOFF** will not alter the global data block logging state (**MAXNDATA|MINDATA|NODATA**).

3) HNAS activates with **TRCVC ON** and **TRCVC MINDATA** set in the absence of the **TRCVC** start parameter.

4) The **TRCVC [ON]**, **TRCVC OFF**, **TRCVC MAXDATA**, **TRCVC MINDATA** and **TRCVC NODATA** start parameters perform the same functions as the **TRCVC ALLON**, **TRCVC ALLOFF**, **TRCVC MAXDATA**, **TRCVC MINDATA** and **TRCVC NODATA** console commands, respectively, for *global* VC event tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

TRCVCQ Command - MCH VC Queue Trace Control (Privileged)

for the VCQ on a specific MCH

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[RNM=*mchname*]	**TRCVCQ**	**[*mchname*]**	**C**	**230**
		[{ON|OFF}] **[DBK {ON|OFF}]**	**C**	**240**

or for a collection of VCQs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**TRCVCQ**	**[{ON|OFF}]** **[DBK {ON|OFF}]**	**C**	**240**
[IFN=*minifn*[-*maxifn*]]				

or for all VCQs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCVCQ	**[{ALLON|ALLOFF}]**	**C**	**230**

This command is used to start (**ON**) or stop (**OFF**) *local* VC event tracing or to initiate *local* data block logging (**DBK**) as new VC connections are established for the MCHs identified by the command modifiers (ID=, IFN=, RNM=). *Local* VC event tracing and *local* data block logging for **all** VCQs can be started (**ALLON**) and stopped (**ALLOFF**). This command is provided so that VC tracing can be initiated as soon as a VCB is added to its active VC queue.

Note: All TRCVCQ arguments can be entered at the same time with each one separated by space character. For example, TRCVCQ ON DBK ON. An error message will be generated if duplicate or conflicting arguments are entered. For example, TRCVCQ ON ON is rejected because ON is duplicated and TRCVCQ ON OFF is rejected because ON and OFF conflict.

Note: VCQ=MCHVCQ for all LLC types (contained in MCH).

Note: For additional information on the meaning of *local* and *global*, please refer to the Terminology section located on page CONS-3 of this document.

- When **RNM=*mchname*** is specified, it overrides the **ID=** and **IFN=** values restricting TRCVCQ processing to the named MCH only. TRCVCQ processing will affect the VCQ on the named MCH only.

- When **IFN=*minifn*[-*maxifn*]** is specified (XTP only), TRCVCQ processing is restricted to the selected MCHs on the selected PCEs. If **IFN=0** is specified, TRCVCQ processing is for all MCHs on the selected PCEs.

- When **ID=*minid*[*-maxid*]** is specified, TRCVCQ processing is restricted to the selected PCEs. If **ID=0** is specified, TRCVCQ processing is for all PCEs. *The ID= modifier is only used if the RNM= modifier is not set. If ID= is also null, the command is rejected.*

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCVCQ Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	OFF	1
OFF	ON	2
ALLON	ALLOFF	3
ALLOFF	ALLON	4
DBK ON	DBK OFF	5
DBK OFF	DBK ON	6

- Enter **TRCVCQ [ON]** to start *local* VCQ event tracing for the VCQs identified by the command modifiers.

- Enter **TRCVCQ OFF** to stop *local* VCQ event tracing and *local* data block logging (disables DBK) for the VCQs identified by the command modifiers.

- Enter **TRCVCQ DBK [ON]** to enable *local* data block logging for the VCQs identified by the command modifiers.

- Enter **TRCVCQ DBK OFF** to disable *local* data block logging for the VCQs identified by the command modifiers.

- Enter **TRCVCQ ALLON** to start *local* VCQ event tracing and *local* data block logging (enables DBK) for **all** VCQs.

- Enter **TRCVCQ ALLOFF** to stop *local* VCQ event tracing and *local* data block logging (disables DBK) for **all** VCQs.

- To summarize, you may enter a TRCVCQ parameter that requests a specific action to be performed as follows:

TRCVCQ Action Matrix

PARAMETER	ACTION
ON or omitted	Activate *local* VCQ event tracing for the VCQs identified by the ID= and IFN= modifiers or the RNM= modifier.

PARAMETER	ACTION
OFF	**Deactivate *local* VCQ event tracing and *local* data block logging for the VCQs identified by the ID= and IFN= modifiers or the RNM= modifier.**
DBK or omitted	**Enable *local* data block logging for the VCQs identified by the ID= and IFN= modifiers or the RNM= modifier.**
DBK OFF	**Disable *local* data block logging for the VCQs identified by the ID= and IFN= modifiers or the RNM= modifier.**
ALLON	**Activate all *local* VCQ event tracing and *local* data block logging (system wide).**
ALLOFF	**Deactivate all *local* VCQ event tracing and *local* data block logging (system wide).**

Example #1: To start *local* VCQ event tracing for all VCQs when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
ID=0 IFN=0           <- force all VCQs to be referenced
TRCVCQ [ON]          <- start local event tracing for all VCQs
```

Example #2: To start *local* VCQ event tracing for all the VCQs associated with a TYPE=MCH|XTP REMOTE when you are not sure of the current state of tracing within HNAS, enter

```
TRCALL STOP          <- stop all tracing
RNM=rmtname          <- identify REMOTE
TRCVCQ [ON]          <- start local event tracing for VCQs on rmtname
```

Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

A description of HNAS VC event trace entries (ID=LSRS|LSCL|...) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCVCQ:

1) HNAS activates with **TRCVCQ OFF** set. There is no equivalent start parameter.

TRCWTO Command - XFWTO Call Trace Control (Privileged)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	TRCWTO	**[{ON|OFF}]**		

This command is used to start (**ON**) or stop (**OFF**) XFWTO subroutine call tracing.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

TRCWTO Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON	**OFF**	**1**
OFF	**ON**	**2**

- XFWTO subroutine call tracing allows alert message requests to be logged in the HNAS internal trace table. This functionality is valuable for remote problem diagnosis when the trace table can be examined by a remote console operator.

  Warning: HNAS tracing requires additional computing cycles which can influence HNAS performance.

  A description of HNAS subroutine call trace entries (ID=80) can be found in the Trace Table Entries section of this product documentation.

General notes for TRCWTO:

1) HNAS activates with **TRCWTO OFF** set in the absence of the **TRCWTO** start parameter.

2) The **TRCWTO [ON]** and **TRCWTO OFF** start parameters perform the same function as the **TRCWTO [ON]** and **TRCWTO OFF** console commands, respectively, for *global* WTO request tracing. For more information on HNAS start parameters, refer to Chapter 2 of the HNAS Guide and Reference documentation.

V or VARY Command - Vary Resource State (Privileged)

(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

for a specific SLU resource

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LUNM=*sluname*	**V[ARY]**	**[*sluname*]**	**N**	**240**
		[LU]	**N**	**240**
		[{ON|ACT|ACTIVE}| **{OFF|INACT|INACTIVE}]** **[{FORCE}]**	**N** **A**	**112** **230**

or for the PCEs associated with a specific REMOTE resource

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
RNM=*rmtname*	**V[ARY]**	**[*rmtname*]**	**C**	**220**
		[RMT]	**C**	**230**
		[{ON|ACT|ACTIVE}| **{OFF|INACT|INACTIVE}]** **[{FORCE}]**	**N** **A**	**112** **230**

or for the PCEs associated with a specific LOCAL resource

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
LNM=*lclname*	**V[ARY]**	**[*lclname*]**	**C**	**230**
		[LCL]	**C**	**230**
		[{ON|ACT|ACTIVE}| **{OFF|INACT|INACTIVE}]** **[{FORCE}]**	**N** **A**	**112** **230**

or for a collection of resource PCEs

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
[ID=*minid*[-*maxid*]]	**V[ARY]**	**[{ON|ACT|ACTIVE}|** **{OFF|INACT|INACTIVE}]** **[{FORCE}]**	**N** **A**	**112** **230**

This command is used to activate (**ON|ACT|ACTIVE**), deactivate (**OFF|INACT|INACTIVE**) or deactivate all (**FORCE**) the HNAS resources (PCEs or LUs) identified by the command modifiers (RNM=, LNM= and LUNM= **are required** for the **RMT** , **LCL** and **LU** parameters, respectively).

Note: The **ON**, **ACT** and **ACTIVE** verbs are equivalents and are interchangeable as are the **OFF**, **INACT** and **INACTIVE** verbs.

Note: The **FORCE** verb implies an unconditional **OFF**. The **OFF** verb may be specified with **FORCE** but is not required.

- When **LUNM=*sluname*** is specified, it overrides the **ID=** values when the LU argument is entered restricting VARY processing to the named SLU only.

- When **RNM=*rmtname*** is specified, it overrides the **ID=** values when the RMT argument is entered (or is defaulted to) restricting VARY processing to the named REMOTE only.

- When **LNM=*lclname*** is specified, it overrides the **ID=** values when the LCL argument is entered restricting VARY processing to the named LOCAL only.

- When **ID=*minid*[-*maxid*]** is specified, VARY processing is restricted to the selected PCEs. A non-zero **ID=** range is required when RNM= and LNM= are omitted. *The ID= modifier is only used if the RNM= and LNM= modifiers are not set. If ID= is also null, the command is rejected.*

  Note: If a resource name (*rmtname*, *lclname* or *sluname*) is supplied as a right side modifier, the corresponding RMT, LCL or LU type argument need not be specified since HNAS can tell by the resource name what type of resource it is. The case when the RMT, LCL or LU argument is required is when permanent modifier values are set so the HNAS knows which one to use for the VARY command. If a right side modifier is supplied with a type argument, an error message will be generate if the resource type does not match the specified type argument. For example, V *sluname* RMT will be rejected.

- Parameters are decoded and remembered and then processed in the **order** listed in the following table.

  VARY Parameter Processing Matrix

PARAMETER	CONFLICT	ORDER
ON|ACT|ACTIVE	OFF|INACT|INACTIVE|FORCE	1
OFF|INACT|INACTIVE|FORCE	ON|ACT|ACTIVE	2

- Enter **V [ON]** to activate resources identified by the given command modifiers if they are currently inactive (offline). No action is taken for resources already in the active state.

- Enter **V OFF** to deactivate resources identified by the given command modifiers if they are currently active (online). No action is taken for resources already in the inactive state.

- Enter **V FORCE** to deactivate resources identified by the given command modifiers if they are currently active and to close all active client sockets associated with these resources. No action is taken for resources already in the inactive state.

- When **V [ON]** is issued for an inactive **TYPE=XOT|XTP LOCAL**, the LOCAL PCE is marked active and the TCP/IP server socket is conditioned to accept inbound connections and allow outbound connections.

- When **V OFF** is issued for an active **TYPE=XOT|XTP LOCAL**, the LOCAL PCE is marked inactive and the TCP/IP server socket is closed. No additional router connections directed at the LOCAL IP address can be accepted, however, existing client connections will remain active until cleared by HNAS or the remote DTE.

- When **V FORCE** is issued for an active **TYPE=XOT|XTP LOCAL**, the LOCAL PCE is marked inactive, the TCP/IP server socket is closed and all active client sockets for which the LOCAL is HOME are also closed. No additional router connections directed at the LOCAL IP address can be accepted.

- When **V [ON]** is issued for an inactive **TYPE=XOT|XTP REMOTE**, the REMOTE PCEs are marked active and all associated TCP/IP client sockets are conditioned to accept inbound connections and allow outbound connections. Outbound connections are allowed only when valid IPADDR= and PORT= operand values are specified.

- When **V OFF** is issued for an active **TYPE=XOT|XTP REMOTE**, the REMOTE PCEs are marked inactive. Active client sockets remain active until cleared by HNAS or the remote DTE.

- When **V FORCE** is issued for an active **TYPE=XOT|XTP REMOTE**, the REMOTE PCEs are marked inactive and all associated TCP/IP client sockets are closed and all VC and LU connections are deactivated. This will result in an UNBIND PIU being passed to the VTAM application and the LU ACB being closed.

- When **V [ON]** is issued for an inactive **TYPE=MCH REMOTE**, the MCH is marked active and all associated SLUs are conditioned to accept inbound connections and allow outbound connections.

- When **V OFF** is issued for an active **TYPE=MCH REMOTE**, the MCH is marked inactive. Active LU/VC sessions remain active until cleared by HNAS or the remote DTE.

- When **V FORCE** is issued for an active **TYPE=MCH REMOTE**, the MCH is marked inactive and all associated LU/VC connections are deactivated. This will result in an UNBIND PIU being passed to the VTAM application and the LU ACB being closed. For **SVCs**, the allocated VC and LU control blocks are released. For **PVCs**, the VC session with the remote remains active (PVC SETUP exchange not required) and the associated LU remains inactive (ACB closed).

- When **V *spuname* [ON]** is issued for an inactive **TYPE=SPU REMOTE** with **OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE** specified, an outbound QLLC call is initiated. If OPTIONS=CLOTINITYP=NONE is in effect, the V ACT command is rejected.

- When **V *spuname* OFF** is issued for an active **TYPE=SPU REMOTE**, the command performs no function. To deactivate an active SPU, you must issue V INACT for the connected TYPE=XOT REMOTE. This can be found using the DPCE command with RNM=*spuname*.

- When **V *sluname* [ON]** is issued for an inactive **SLU**, the SLU is marked active and is conditioned to accept inbound connections and allow outbound connections. If the SLU is associated with a **PVC** session the LU ACB is opened only if the PVC session with the router has been created by XOT PVC Setup packets. The **status** of a PVC LU or VC can be displayed with the HNAS **DLU** or **DVC** console commands. The **LUOPT** column shows if the LU ACB is open, the **VCOPT** column shows if the VC has a network connection (PVC SETUP sequence complete).

- When **V *sluname* OFF** is issued for an active **SLU**, the SLU is marked inactive. An active LU/VC session remains active until cleared by HNAS or the remote DTE. The VC control block is left intact (socket not closed, TCP/IP VC session not torn town). For a PVC, a SETUP exchange is not required because the setup state is still active.

- When **V *sluname* OFF FORCE** is issued for an inactive **SLU**, the SLU is marked inactive and the TCP/IP socket is closed for the SLU if a socket is active (normally, only a PVC can be in this state). This logic was implemented by APAR 2400098. Prior to APAR 2400098, the VARY *sluname* FORCE command would have been rejected when issued to an already inactive SLU. The new logic ensures that the TCP/IP socket connection is torn down regardless of the state of the SLU.

  Note that this new logic eliminates the two step process of first finding the PID for the VC socket (using the DLU command) and then issuing the VARY ID=pid FORCE command to close the VC socket. This new support allows customers to work with an *sluname* which is useful for installations that employ automated console command processes.

  PVC SETUP note: Closure of the socket causes the XOT PVC to re-enter PVC SETUP pending state which will cause the PVC SETUP to be sent from the appropriate initiator once the retry timer expires.

- When **V *sluname* OFF FORCE** is issued for an active **SVC SLU**, the SLU is marked inactive and any associated LU/VC connection is deactivated. This will result in an UNBIND PIU being passed to the VTAM application and the LU ACB being closed. The allocated VC and LU control blocks are released and the associated TCP/IP socket is closed for the SLU.

- When **V *sluname* OFF FORCE** is issued for an active **PVC SLU**, the associated LU connection is deactivated and the associated LU/VC connection is deactivated. An UNBIND PIU is passed to the PLU and the LU's ACB is closed and the associated TCP/IP socket is closed for the SLU.

- When **V ID=*pid* FORCE** is issued for an active **PVC** PCE (the ID=*pid* value is displayed under the RPID column of the DVC display), the associated LU/VC connection is deactivated. This will result in an UNBIND PIU being passed to the VTAM application and the LU ACB

being closed. The allocated VC and LU control blocks are then refreshed. This also conditions the **PVC** to either send or await a PVC SETUP from the remote router.

Note: When a resource (REMOTE, LOCAL or SLU) is varied offline via a V OFF [FORCE] command, it will remain unavailable for subsequent use until it is varied back online via the V ON command.

Note: The FORCE parameter is only used in conjunction with the OFF (=INACT) parameter and only for TYPE=XOT|XTP|MCH REMOTE resources, TYPE=XOT|XTP LOCAL resources or SLUs. The FORCE parameter is ignored when ON (=ACT) is specified or when a TYPE=SPU REMOTE is specified. If FORCE is entered without OFF or INACT, OFF is assumed.

Note: HNAS activates with the REMOTE and LOCAL resource state set based on the INIT= operand of the REMOTE and LOCAL definition statements. If INIT=IDLE is specified, the resource is left in the inactive (offline) state when HNAS starts. The **V ON** console command is then required to activate the resource. If INIT=ACTIVE is specified or if the INIT= parameter is omitted (ACTIVE is the default), the resource is automatically activated (brought online). For more information on the INIT= operand of the REMOTE and LOCAL definition statements, refer to Chapter 4 of the HNAS Guide and Reference documentation.

Note: Varying a LOCAL resource OFF and then ON again in rapid succession can cause a TCPIP BIND request to fail and the following alarm message to be displayed:

```
NAS2321W BIND REQUEST FAILED RC=FFFFFFFF 00000030
```

The ERNO=30 at the end of the RC indicates that LOCAL ‘IP address is in a timed wait because of a LINGER delay from a previous close or another process is using the address’. When this event occurs, HNAS enforces the DELAYTIME= value and will not retry the BIND until the delay timeout expires. At this time, HNAS will also reset the LINGER timeout that the stack is enforcing thus allowing the BIND to complete. If the DELAYTIME= value is too high, it can take a long time for a BIND retry. To avoid long delays, we recommend that you specify DELAYTIME=1 in the CDF or modify the DELAYTIME= value using the MLCL INIT=DELAYTIME=1 console command.

VCN= Modifier - Set VCB Identifier

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
VCN=*minvcn*[-*maxvcn*] **VCN=?**	***command***	**[VCN=*minvcn*[-*maxvcn*]]**	**C**	**240**

This modifier is used to select specific VCB(s) for the commands that follow. HNAS allocates a VCB for each REMOTE SVC and PVC connection. The **VCN=** value serves as an identifier for the VC resource. For more information on VCB identifiers, refer to page CONS-37 of this document.

Note: The VCN= modifier is used by the DVC and TRCVC commands.

- If VCN= (null) is specified, the command operates on all VCBs unless noted otherwise.
- If VCN=0 is specified (VCN=0-0 is not permitted), the command also operates on all VCBs unless noted otherwise. For all commands, VCN= (null) and VCN=0 are treated identically.
- If a VCN= value is specified as a range (e.g., VCN=3-10), the command operates on the selected VCBs.
- If a VCN= value is specified as a single element (e.g., VCN=3) or as a range where the extents are the same (e.g., VCN=3-3), the command operates on the selected single VCB.
- If an invalid VCN= value is specified, the balance of the command line is ignored.
- If a VCN= value is specified, it must be entered as a decimal number.

  VCN=*minvcn*[-*maxvcn*] values can range from 1 to 4095.

- Enter VCN=? to display the permanent VCN= value.

WRU Command - Display NASNAME= Operand (Who Are You)

(new for V2R3M0)

Prefix Modifiers	Command	Suffix Modifiers, Parameters	CC	VRM
	WRU		**N**	**230**

This command will display the value specified for the NASNAME= operand on the BUILD definition statement.

The following display is produced when **WRU** is entered.

I AM *nasname*

Console Command Error Messages

The HNAS console subsystem produces diagnostic error messages during processing of some console commands. The message identifier for each message begins with the string “NASC”. The error messages are generated for the local SYSCONS (master operator console) as well as remote consoles. All messages have the following basic format:

NAS	*cdrms*		Description
NAS			**is the Host NAS console message identifier.**
	c		**is a component identifier.**
		C	indicates a console related message.
	d		**is a definition statement identifier.**
		0	indicates a problem unrelated to the type of resource.
		1	indicates a problem while processing a BUILD resource.
		2	indicates a problem while processing a LOCAL resource.
		3	indicates a problem while processing a REMOTE resource.
		4	indicates a problem while processing an SLU resource.
		5	indicates a problem while processing a command argument.
	r		**is a reason code.**
		0	indicates that an operand value was omitted.
		1	indicates that an operand value was invalid or in conflict.
		2	indicates that an operand value was in error.
		>2	indicates a miscellaneous error or informational message.
	m		**is a message number.**
	s		**is a severity code.**
		I	indicates information only.
		D	indicates a default action was taken.
		W	indicates a warning.
		E	indicates an error.

HNAS console error messages maintain the same format as HNAS configuration and alert messages. However, unlike HNAS configuration and alert messages, console error mes-

sages will not affect HNAS operation. Console error messages are generated solely to indicate why a command cannot be completely or partially executed.

NASC002E *cmdname* COMMAND REJECTED, PRIVILEGED

EXPLANATION:

The ***cmdname*** console command cannot be executed because it is a privileged command and the remote console operator does not have privileged status.

OPERATOR ACTION:

Logoff (Quit) the remote console session then logon again entering the console password *backwards* to obtain privileged status.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC003E DECODE FAILURE: *badcmd...*, RE-ENTER

EXPLANATION:

The console command ***badcmd*** contains a spelling or syntax error. This can also occur if a non-printable character was inadvertently entered.

OPERATOR ACTION:

Correct the error then re-enter the command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

Note: Many commands accept a resource name following the command. For example, DRMT *rmtname* SVC0. If *rmtname* is spelled incorrectly or if it is not defined in CDF on a REMOTE definition statement, the command may be rejected for the lack of a valid RNM= or ID= modifier which will cause the *rmtname* to be processed as the *badcmd*. This is because commands that are rejected for a missing or erroneous LUNM=, LNM= RNM=, or ID= modifier, return control to the console input parser which will continue processing the current input record from the end of the rejected command. In the example above, this would be after DRMT which means that *rmtname* would be processed as if it were a new command resulting in the NASC003E message above.

NASC005W *cmdname* ABORTED DUE TO OPERATOR INPUT

(new for V2R4M0)

EXPLANATION:

During the execution of the console command ***cmdname***, input was entered that terminated the current command. Input entered during any command is interpreted as an attention signal that pre-empts the current command thus allowing a new command to be executed.

OPERATOR ACTION:

None.

SYSTEM ACTION:

The input is processed as a new command.

NASC006W *cmdlist* COMMAND LIST INTERRUPTED

(new for V2R4M0)

EXPLANATION:

During the execution of the console command list ***cmdlist*** (e.g. **EXEC *ddname***), input was entered that terminated the current command within the *cmdlist*. Input entered during any command is interpreted as an attention signal that pre-empts the current command thus allowing a new command to be executed.

OPERATOR ACTION:

None.

SYSTEM ACTION:

The input is processed as a new command which interrupts but does not abort the current *cmdlist*.

NASC013E UNSUPPORTED FUNCTION, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command requested an operation that could not be performed or is not supported.

OPERATOR ACTION:

Correct the error then re-enter the command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC014E FUNCTION ALREADY ACTIVE, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command requested an operation that is supposed to activate a component but the component was already active when the command was issued.

OPERATOR ACTION:

None. The request is ignored.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC015E FUNCTION ALREADY INACTIVE, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command requested an operation that is supposed to deactivate a component but the component was already inactive when the command was issued.

OPERATOR ACTION:

None. The request is ignored.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC020W RESOURCE BUSY, RETRY *cmdname* COMMAND LATER

EXPLANATION:

The ***cmdname*** console command was issued by two (2) different console operators and both reference the same resource (e.g., **MRMT *mchname* LUNAME=*sluname*/...**). Only one ***cmdname*** command at a time is allowed to operate on a resource (first come, first served).

OPERATOR ACTION:

Wait a moment then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC021W RESOURCE UNAVAILABLE, RETRY *cmdname* COMMAND LATER

EXPLANATION:

The ***cmdname*** console command requires a resource such as a buffer or TCP/IP PCE for its execution (e.g., **PING *ipaddr***) but none were available.

OPERATOR ACTION:

Wait a moment then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC022W RESOURCE ONLINE, VARY OFFLINE THEN RETRY *cmdname* COMMAND

(new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command requires a resource such as TYPE=XOT REMOTE to be offline for its execution (e.g., **MRMT *rmtname* IPADDR=*ipaddr***) but the REMOTE was currently online.

OPERATOR ACTION:

Issue the **VARY *rmtname* FORCE** command to force the REMOTE offline then retry the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC023W RESOURCE OFFLINE, VARY ONLINE THEN RETRY *cmdname* COMMAND

(new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command requires a resource such as TYPE=XOT REMOTE to be online for its execution but the REMOTE was currently offline.

OPERATOR ACTION:

Issue the **VARY *rmtname* ON** command to force the REMOTE online then retry the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC030E *opname opval* HAS INVALID FORMAT, *cmdname* COMMAND ABORTED

(apar/new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command specified an operand ***opname*** whose value ***opval*** has an invalid format or is inconsistent for the ***opname*** operand. For example, if **MRMT *rmtname* LOGTAB=ISTINCDT** is entered, this message will be issued because ISTINCDT is a standard USSTAB not a LOGATB.

OPERATOR ACTION:

Re-enter the ***cmdname*** command with a valid ***opval*** for the ***opname***.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC031E *opname opval* IS CURRENTLY IN USE, RETRY *cmdname* COMMAND LATER

(apar/new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command specified an operand ***opname*** whose value ***opval*** is currently being used so it cannot be changed or update. For example, if **MRMT *rmtname* USSTAB=ISTINCDT-R** is entered to force ISTINCDT to be reloaded, this message

will be if a USSMSG within ISTINCDT is currently being transmitted. The USSTAB cannot be reloaded if a USSMSG transmission is in progress because this could cause an addressing exception ABEND if establish USSMSG addresses are changed during the transmission.

OPERATOR ACTION:

Wait a moment then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC050E EXEC *ddname* INVALID, CANNOT POINT AT ITSELF, EXEC COMMAND ABORTED

(new for V2R4M0)

EXPLANATION:

The ***ddname*** specified for the EXEC console command identifies a file that contains the same **EXEC *ddname*** specification. This would create an infinite command loop which is not permitted.

OPERATOR ACTION:

Remove the **EXEC *ddname*** statement from the specified file then re-enter the command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC051E *cmdname ddname* INVALID, CANNOT BE OPENED, *cmdname* COMMAND ABORTED

(new for V2R4M0)

EXPLANATION:

The ***ddname*** specified for the ***cmdname*** console command (EXEC or ECHEDULE) identifies a DDNAME that does not exist in the HNAS start JCL which prevents the associated DCB from being opened. This is most likely due to a spelling error.

OPERATOR ACTION:

Enter a valid ***ddname*** that exists in the HNAS start JCL for the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

Note: This connsole warning message was changed after the initial 240 release by APAR 2400064.

NASC052W EXEC COMMAND LIST WAS NOT PROVIDED, EXEC COMMAND ABORTED

(apar/new for V2R4M0)

EXPLANATION:

The ***ddname*** was omitted for the EXEC LIST console command which means there is no command list to display.

OPERATOR ACTION:

Enter a valid ***ddname*** that exists in the HNAS start JCL for the EXEC LIST command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

Note: This connsole warning message was changed after the initial 240 release by APAR 2400064.

NASC053E EXEC *ddname* FILE IS NULL, EXEC COMMAND ABORTED

(apar/new for V2R4M0)

EXPLANATION:

A ***ddname*** was specified for the EXEC console command but the referenced command list file is empty (contains no commands or display comments (# text) but may contain non-display comments ('* text')).

OPERATOR ACTION:

Enter a valid ***ddname*** that identifies a non-empty command list file for the EXEC command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC054E ***cmdname ddname*** **COMMAND QUEUE LIMIT** ***count/limit*** **EXCEEDED,** ***cmdname*** **COMMAND ABORTED**

(new for V2R4M0)

EXPLANATION:

The ***ddname*** specified for the ***cmdname*** console command (EXEC or ECHEDULE) identifies a file that contains more command data than can be queued for execution. Currently ***limit*** is set to 512 bytes. ***count*** is the number of bytes that is required to hold the ***cmdname*** file data.

For the **EXEC** command, assume ***ddname*** identifies a file containing the following statements:

```
DNAS APAR                       <- 9 characters
DPCE TYPE=CONS                  <- 14 characters
DVC                             <- 3 characters
```

For this file, the internal console command queue would look as follows:

```
001D                            <- 29 total bytes
09                              <- first command length
C4D5C1E240C1D7C1D9              <- DNAS APAR
0E                              <- second command length
C4D7C3C540E3E8D7C57EC3D6D5E2    <- DPCE TYPE=CONS
03                              <- third command length
C4E5C3                          <- DVC
```

For the **SCHEDULE** command, assume ***ddname*** identifies a file containing the following statements:

```
(12:00:00,ALARM LOG=?)          <- time=6, command=11 characters
(06:00:00,TRCPCE ALLON)         <- time=6, command=12 characters
(07:00:00,TRCPCE ALLOFF)        <- time=6, command=13 characters
```

For this file, the internal console command queue would look as follows:

```
0039                            <- 57 total bytes
120000120005                    <- first command time
0B                              <- first command length
C1D3C1D9D440D3D6C77E6F          <- ALARM LOG=?
060000060005                    <- second command time
0C                              <- second command length
E3D9C3D7C3C540C1D3D3D6D5        <- TRCPCE ALLON
070000070005                    <- third command time
0D                              <- third command length
E3D9C3D7C3C540C1D3D3D6C6C6      <- TRCPCE ALLOFF
```

OPERATOR ACTION:

The ***cmdname*** command list cannot be executed. Reduce the number of commands in the command list file then re-enter the command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

Note: This connsole warning message was changed after the initial 240 release by APAR 2400064.

NASC055I *cmdname ddname* COMMAND LIST HAS BEEN PROCESSED, IT WILL NOW BE *action*

(new for V2R4M0)

EXPLANATION:

The ***ddname*** specified for the ***cmdname*** console command (EXEC or ECHEDULE) identifies a valid command list file. The file has been read and the specified ***action*** (EXECUTED or LISTED) will be taken.

OPERATOR ACTION:

None.

SYSTEM ACTION:

The ***cmdname*** command list will now run.

Note: This connsole warning message was changed after the initial 240 release by APAR 2400064.

NASC090E NEWDEFN ARRAY IS FULL, MAXIMUM RECORD COUNT OF *ddddd* HAS BEEN REACHED

EXPLANATION:

The console command (e.g., **MLCL** or **MRMT**) attempted to update the NEWDEFN array because of an addition to a LOCAL or REMOTE operand but no available NEWDEFN records were available. HNAS allocates the NEWDEFN array after the CDF is scanned based on the number of records in the CDF. Currently, the NEWDEFN record count is fixed at 3 times the number of records in the CDF (***ddddd***).

OPERATOR ACTION:

None. Contact Comm-Pro for assistance.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC091E NEWDEFN ARRAY LOOKUP FAILED, PARMLIST FOLLOWS
REMOTE *rmtname opname xx...xx*

EXPLANATION:

The console command (e.g., **MLCL** or **MRMT**) attempted to update the NEWDEFN array because of an addition, deletion or change to a LOCAL or REMOTE operand but the NEWDEFN update routine could not resolve the operand in the CDF. This is a severe error.

OPERATOR ACTION:

None. Contact Comm-Pro for assistance.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC092E SEARCH LIST OMITTED, REQUIRED FOR *cmdname* EXECUTION

EXPLANATION:

The ***cmdname*** console command (e.g., **DNWDF**) requires a search list but none was generated during configuration processing.

OPERATOR ACTION:

The ***cmdname*** command cannot be executed.

For DNWDF processing, stop then restart HNAS with GENNWDF specified as a start parameter and //NEWDEFN as a DDNAME.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC093E SEARCH ARGUMENT NOT FOUND, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command processor was given a lookup argument (e.g., **DNAS APAR *aparid***) but the argument could not be found.

OPERATOR ACTION:

Enter a valid search argument then retry the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC100E ID= OMITTED, REQUIRED FOR *cmdname* EXECUTION

(added for V2R3M0)

EXPLANATION:

The ***cmdname*** console command (e.g., **TRCLU**) requires the LUNM=, RNM= or ID= modifier but all were omitted. The ID= modifier is only used for TRCLU when LUNM= and RNM= are not set. If ID= is also omitted, the command cannot be executed.

OPERATOR ACTION:

Supply a valid LUNM=, RNM=, LNM= or ID= value (as appropriate) then re-enter the ***cmdname*** command.

Note: For most commands, ID= omitted and ID=0 are treated identically. Exceptions are for TRCLUQ, TRCMCH, TRCPCE, TRCVC, TRCVCQ, VARY RMT and MON TAP where an ID= value must be supplied when RNM= is omitted, for TRCLU and TRCMCHX where an ID= value must be supplied when RNM= and LUNM= are omitted and for VARY LCL where an ID= value must be supplied when LNM= is omitted.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC100E ID= AND OTHER MODIFIERS OMITTED, REQUIRED FOR *cmdname* EXECUTION
NASC100E ID= IS REQUIRED WHEN NO OTHER RESOURCE IDENTIFICATION IS SPECIFIED
NASC100E IF COMMAND OPERATES ON LNM=, RNM= OR LUNM=, ENSURE THAT RESOURCE NAME
NASC100E IS SPECIFIED BEFORE ANY COMMAND OPERANDS (EXAMPLE: V LUNM=*sluname* OFF)

(apar/change for V2R4M0)

EXPLANATION:

The ***cmdname*** console command (e.g., **VARY**) requires the LUNM=, RNM=, LNM= or ID= modifier but all were omitted. The ID= modifier is only used for VARY when LUNM=, RNM= and LNM= are not set. If ID= is also omitted, the command cannot be executed.

OPERATOR ACTION:

Supply a valid LUNM=, RNM=, LNM= or ID= value (as appropriate) then re-enter the ***cmdname*** command.

Note: For most commands, ID= omitted and ID=0 are treated identically. Exceptions are for TRCLUQ, TRCMCH, TRCPCE, TRCVC, TRCVCQ, VARY RMT and MON TAP where an ID= value must be supplied when RNM= is omitted, for TRCLU and TRCMCHX where an ID= value must be supplied when RNM= and LUNM= are omitted and for VARY LCL where an ID= value must be supplied when LNM= is omitted.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC110E ID=*lo-hi* INVALID, *cmdname* COMMAND ABORTED

(added for V2R4M0)

EXPLANATION:

The ***cmdname*** console command requires an ID= value but the specified value is not valid because is exceeds the maximum configured PCEID value or the given *lo-hi* range identifies PCEs that are invalid for the given command.

OPERATOR ACTION:

Supply a valid ID= value then re-enter the ***cmdname*** command.

Note: For most commands, ID= omitted and ID=0 are treated identically. Exceptions are for TRCLUQ, TRCMCH, TRCPCE, TRCVC, TRCVCQ, VARY RMT and MON TAP where an ID= value must be supplied when RNM= is omitted, for TRCLU and TRCMCHX where an ID= value must be supplied when RNM= and LUNM= are omitted and for VARY LCL where an ID= value must be supplied when LNM= is omitted.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC121E DDNAME *ddname* CANNOT BE OPENED, CHECK HNAS START JCL

EXPLANATION:

The TRCTRAP console command processor was given a TRAPACTION argument of **EXEC=*ddname*** but the specified ***ddname*** could not be opened.

OPERATOR ACTION:

Enter a valid ***ddname*** that exists in the HNAS start JCL for the EXEC command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC200E LNM= OMITTED, REQUIRED FOR *cmdname* EXECUTION

EXPLANATION:

A LOCAL definition statement name is required for the ***cmdname*** console command (e.g., **MLCL**) but none was not provided.

OPERATOR ACTION:

Supply a valid LOCAL definition statement name using the LNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC204E *ddname* ABEND PARMLIST=*xxxxxxxx ... xxxxxxxx*

(new for V2R4M0)

EXPLANATION:

The ***ddname*** specified for the EXEC console command or TRAPACTION=EXEC= operand identifies a file that cannot be read. For example, it points at a member of a partitioned dataset that does not exist. The PARMLIST value provides the ABEND code and reason.

OPERATOR ACTION:

The EXEC command list cannot be executed. Correct the error then re-enter the command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC210E LNM=*lclname* INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***lclname*** specified for the LNM= modifier for the ***cmdname*** console command does not identify a LOCAL definition statement. This is most likely due to a spelling error.

OPERATOR ACTION:

Supply a valid LOCAL definition statement name using the LNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC211E LNM=*lclname* TYPE=*type* INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

A valid LOCAL definition statement name was provided for the ***cmdname*** console command (e.g., **MLCL**) but it identifies a LOCAL having the wrong ***type*** (e.g., TYPE=XTP when TYPE=XOT is required).

OPERATOR ACTION:

Supply a LOCAL definition statement name having the correct ***type*** using the LNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC212E LNM=*lclname* CONFIG INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

A valid LOCAL definition statement name was provided for the ***cmdname*** console command (e.g., **MLCL**) but it identifies a LOCAL having a configuration error (e.g., RTE-OUT= operand is required but was not specified).

OPERATOR ACTION:

Supply a LOCAL definition statement name that has the correct configuration operands using the LNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC300E RNM= OMITTED, REQUIRED FOR *cmdname* EXECUTION

EXPLANATION:

A REMOTE definition statement name is required for the ***cmdname*** console command (e.g., **MRMT**) but none was not provided.

OPERATOR ACTION:

Supply a valid REMOTE definition statement name using the RNM= modifier then re-enter the ***cmdname*** command.

Note: Console commands that require RNM= have always been rejected with the NASC300E message. This message will now also be issued for a command that requires either RNM= or ID= and both are omitted. For example, MON TAP requires either RNM=*rmtname* or ID=*lo*[-*hi*] to be provided. If both RNM= and ID= are omitted (not initialized), the command will be rejected and the NASC300E message will be issued.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC310E RNM=*rmtname* INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***rmtname*** specified for the RNM= modifier for the ***cmdname*** console command does not identify a REMOTE definition statement. This is most likely due to a spelling error.

OPERATOR ACTION:

Supply a valid REMOTE definition statement name using the RNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC311E RNM=*rmtname* TYPE=*type* INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

A valid REMOTE definition statement name was provided for the ***cmdname*** console command (e.g., **MRMT**) but it identifies a REMOTE having the wrong ***type*** (e.g., TYPE=XTP when TYPE=MCH is required).

OPERATOR ACTION:

Supply a REMOTE definition statement name having the correct ***type*** using the RNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC312E RNM=*rmtname* CONFIG INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

A valid REMOTE definition statement name was provided for the ***cmdname*** console command (e.g., **MRMT**) but it identifies a REMOTE having a configuration error (e.g., SVC5= is operand required but was not specified).

OPERATOR ACTION:

Supply a REMOTE definition statement name that has the correct configuration operands using the RNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC313W REMOTE *rmtname* NOT FOUND, *cmdname* COMMAND ABORTED

EXPLANATION:

A valid REMOTE definition statement name was specified for an operand entry (LUNAME=, SVC0=, SVC3= or SVC5=) but the corresponding REMOTE definition statement named ***rmtname*** could not be found in the CDF.

OPERATOR ACTION:

Supply a REMOTE definition statement name that represents an MXT then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC313W REMOTE *rmtname* IS NOT TYPE=MXT, *cmdname* COMMAND ABORTED

EXPLANATION:

A valid REMOTE definition statement name was specified for an operand entry (LUNAME=, SVC0=, SVC3= or SVC5=) but the corresponding REMOTE definition statement named ***rmtname*** was not TYPE=MXT.

OPERATOR ACTION:

Supply a REMOTE definition statement name that represents an MXT then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC400E LUNM= OMITTED, REQUIRED FOR *cmdname* EXECUTION

EXPLANATION:

An LU name is required for the ***cmdname*** console command (e.g., **MRMT**) but one was not provided.

OPERATOR ACTION:

Supply a valid LU name using the LUNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC410E LUNM=*sluname* INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***sluname*** specified for the LUNM= modifier for the ***cmdname*** console command does not identify an LU resource. This is most likely due to a spelling error.

OPERATOR ACTION:

Supply a valid LU name using the LUNM= modifier then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC421E REMOTE SLU *sluname* IS NOT IDLE, MRMT COMMAND ABORTED

EXPLANATION:

An **MRMT** request to modify an operand entry (LUNAME=, SVC0=, SVC4= or SVC5=) of a TYPE=MCH|XTP|SPU REMOTE definition statement referenced an SLU named ***sluname*** that was active which precludes changing its configuration at the current time. The SLU must be inactive in order for it to be removed or renamed.

OPERATOR ACTION:

Force the SLU inactive using the VTAM **VARY NET,INACT,ID=*sluname*** command then retry the **MRMT** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC422E REMOTE SLU *sluname* ALREADY EXISTS, MRMT COMMAND ABORTED

EXPLANATION:

An **MRMT** request to add an operand entry (LUNAME=, SVC0=, SVC4= or SVC5=) of a TYPE=MCH|XTP|SPU REMOTE definition statement referenced an SLU named ***sluname*** that already exists and thus cannot be added again. The new SLU name must be unique.

OPERATOR ACTION:

Supply a unique SLU name then re-enter the **MRMT** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC423E REMOTE SLU *sluname* COULD NOT BE ALLOCATED, MRMT COMMAND ABORTED

EXPLANATION:

An **MRMT** request to add an operand entry (LUNAME=, SVC0=, SVC4= or SVC5=) of a TYPE=MCH|XTP|SPU REMOTE definition statement could not be satisfied because all SLUs in the LU Dynamic Reconfiguration Pool (LUDRPOOL) were already allocated.

OPERATOR ACTION:

None. The command cannot be executed.

Increase the number of free SLUs in the LUDRPOOL using the LUDRPOOLCNT= suboperand of the OPTIONS operand on the BUILD definition statement. This will require that HNAS be shutdown and restarted.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC500E DEFAULT PARAMETER OMITTED, REQUIRED FOR *cmdname* EXECUTION

EXPLANATION:

The ***cmdname*** console command (e.g., **PING**) requires default parameter value(s) when no arguments are specified but none were provided using the command modifiers.

OPERATOR ACTION:

Enter default values using the command modifiers or as arguments for the ***cmdname*** command then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC511E PARAMETERS CONFLICT, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command requested an operation that could not be performed based on the current resource configuration (e.g., **VARY *spuname* ACT** issued but OPTIONS=CLOTINITYP=NONE is in affect for the named SPU) or conflicting or mutually exclusive arguments were specified (e.g., **PRNT ON OFF**).

OPERATOR ACTION:

Ensure that the ***cmdname*** command is correct for the configuration (set OPTIONS=CLOTINITYP=CONSOLE using the MRMT command for the VARY command) and ensure that mutually exclusive arguments are not specified then retry the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC512E PARAMETERS DUPLICATED, *cmdname* COMMAND ABORTED

(new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command argument has been specified more than once for the command (e.g., **PRNT ON ON**).

OPERATOR ACTION:

Re-enter the ***cmdname*** command with the correct arguments specified.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC513E PARAMETER LIMIT EXCEEDED, *cmdname* COMMAND ABORTED

(new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command list argument has been specified that causes an array to reach it's limit which cannot be allowed (e.g., **PRNT SWITCH=(*ddnmlist*)**).

OPERATOR ACTION:

Re-enter the ***cmdname*** command with fewer elements specified or reset the array then re-enter the list (e.g., **PRNT SWITCH=* PRNT SWITCH=(*ddnmlist*)**).

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC515E STATE CONFLICT, ISSUE THE RESUME FUNCTION THEN RETRY THE *cmdname* COMMAND

(new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command action could not be performed due to the state of the target resource or configuration parameter (e.g., **PRNT NEXTPRSW** was specified when the PRTSWLST= operand on the BUILD definition statement was in the STOPped state (last DDNAME had been used)).

OPERATOR ACTION:

Enter the resume function for the ***cmdname*** command that issued this message (e.g., enter **PRNT RSMEPRSW**).

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC520E LIST ENTRY NAME INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command list form argument specifies a resource name (e.g., **MRMT SVC0=*sluname***) that contains a spelling, syntax or length error. This could also occur if a non-printable character was inadvertently entered.

OPERATOR ACTION:

Supply a valid resource name then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC521E LIST ENTRY INDEX INVALID, *cmdname* COMMAND ABORTED

EXPLANATION:

The ***cmdname*** console command list form argument specifies an entry index (e.g., **MRMT *rmtname* LUNAME=(*locaddr*,...)**) that is invalid or too large for the list form configuration operand.

OPERATOR ACTION:

Supply a valid list from operand index then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC531E PARAMETER DATA OMITTED, REQUIRED FOR *cmdname* COMMAND EXECUTION

(new for V2R4M0)

EXPLANATION:

The ***cmdname*** console command argument is required but was not provided.

OPERATOR ACTION:

Re-enter the ***cmdname*** command with the required argument.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC532E PARAMETER DATA INVALID: *baddata, cmdname* COMMAND ABORTED

(changed for V2R4M0)

EXPLANATION:

The ***cmdname*** console command specifies a parameter that is invalid due to a spelling, value or syntax error. ***baddata*** represents up to 8 bytes before and after the parameter data that is in error.

OPERATOR ACTION:

Correct the parameter in error then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC533E *ddname* FILE DATA INVALID: *baddata, cmdname* COMMAND ABORTED

(added for V2R4M0)

EXPLANATION:

The ***ddname*** file specified for the ***cmdname*** console command (EXEC or SCHEDULE) identifies a file that contains a record with invalid data due to a spelling, value or syntax error. ***baddata*** represents up to 8 bytes before and after the parameter data that is in error.

OPERATOR ACTION:

Correct the ***ddname*** file parameter data in error then re-enter the ***cmdname*** command.

SYSTEM ACTION:

Additional parsing of the current ***ddname*** file is terminated and the console input prompt message is issued to solicit a new command.

NASC611W DISPLAY MEMORY ADDRESSING EXCEPTION,
DMEM COMMAND ABORTED

EXPLANATION:

The **DMEM** console command referenced an address that HNAS is not authorized to display (address is invalid for machine configuration).

OPERATOR ACTION:

Correct the memory address argument and re-enter the **DMEM** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC611W FIND MEMORY ADDRESSING EXCEPTION,
FIND COMMAND ABORTED

EXPLANATION:

The **FIND** console command referenced an address that HNAS is not authorized to display (address is invalid for machine configuration).

OPERATOR ACTION:

Correct the memory address argument and re-enter the **FIND** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

NASC611W MODIFY MEMORY ADDRESSING EXCEPTION, MMEM COMMAND ABORTED

EXPLANATION:

The **MMEM** console command referenced an address that HNAS is not authorized to modify (address is invalid for machine configuration or address is not in the same storage key as HNAS).

OPERATOR ACTION:

Correct the memory address argument and re-enter the **MMEM** command.

SYSTEM ACTION:

Additional parsing of the current input record is terminated and the console input prompt message is issued to solicit a new command.

Trace Entry Formats

(Trace Table Entries)

This page left intentionally blank.

Introduction

HNAS trace entries are logged in a wrap around trace table within the HNAS address space. Each trace entry comprises sixteen (16) contiguous bytes within the trace table. Multiple 16-byte entries are normally used for each event that is traced. The size of the trace table is determined from the TRCLMT operand of the BUILD definition statement.

In HNAS V2R2M0 TRCLMT=4000 (4000 trace entries) is now the default which should accommodate general tracing requirements. We suggest that you code TRCLMT=8000 should their be a requirement to run extended or global tracing.

In addition to the trace table, trace entries can also be written to the HNAS SYSPRINT console log. This is accomplished through the use of the of the TRCPRNT start parameter or via the TRCPRNT ON console command.

For more information on the TRCPRNT start parameter, refer to Chapter 2 of the HNAS Guide and Reference documentation.

For more information on the TRCPRNT console command, refer to Console section of the HNAS Console Subsystem and Operations Guide.

Trace Entry Identifiers: Quick Reference

ID	TRACE EVENT	REFERENCE	CC	VRM
0	CPU Utilization	Page TRC-4		
1	Subtask Dispatch	Page TRC-5		
2	Timer Completion	Page TRC-6		
3	TCP/IP I/O Request	Page TRC-14		
4	TCP/IP External Interrupt	Page TRC-16		
5	TCP/IP Request Buffer	Page TRC-18		
6	TCP/IP Response Buffer	Page TRC-21		
7	TCP/IP I/O Data	Page TRC-25		
80	Subroutine Call	Page TRC-26		
81	Binary Search Call	Page TRC-26		
9	Console Subtask Request	Page TRC-26		
A	TCP/IP RCV Completion	Page TRC-7		
B	TCP/IP XMT Completion	Page TRC-8		
C	TCP/IP Subtask Completion	Page TRC-11		
E	Main Task WAIT	Page TRC-11		210
F	Main Task POST	Page TRC-11		210
DBLK	Block Dump Start	Page TRC-31		
DBLK-END	Block Dump End	Page TRC-31		
BFCM	Buffer Completion	Page TRC-34		
BFR	Buffer Data	Page TRC-34		
GDIA	GATE Diagnostic Message	Page TRC-32		
LSCL	VC Clear Request Scheduled	Page TRC-36		
LSRS	VC Reset Request Scheduled	Page TRC-35		
LURL	LU Releasing Resources	Page TRC-64		
LUSE	LU MCH Input State Error	Page TRC-65		
LUTO	LU Input Queue Timeout	Page TRC-63		
L3RQ/S	LLC3 Req or Rsp Received	Page TRC-66	A	220
ROCR	XOT VC RCV Call Request	Page TRC-44		

ID	TRACE EVENT	REFERENCE	CC	VRM
ROCR-FAIL	XOT VC RCV Call Failure	Page TRC-45		
ROdd	XOT VC RCV Data	Page TRC-46		
RTCR	XTP VC RCV Call Request	Page TRC-40		
RTCR-FAIL	XTP VC RCV Call Failure	Page TRC-41		
RTdd	XTP VC RCV Data	Page TRC-42		
RTMC	XTP MCH RCV Data	Page TRC-39		
VCRL	VC Releasing Resources	Page TRC-37		
VCTO	VC Timeout	Page TRC-38		
XBt	XMT Buffer Request	Page TRC-33		
Xpdd	VC XMT Data	Page TRC-48		
XpXC	VC XMT Completion	Page TRC-50		
VCLO	LU Close ACB	Page TRC-52		
VOPN	LU Open ACB	Page TRC-51	C	210
VRC	LU RCV Completion	Page TRC-54		
VRF	LU RCV Flush	Page TRC-59		
VRIt	LU RCV Issued	Page TRC-53	C	210
VRN	LU RCV Notify Request	Page TRC-57		
VRTE	LU RCV TPOPEN Request	Page TRC-58		
VRX	LU RCV Expedited Request	Page TRC-56		
VSC	LU SEND Completion	Page TRC-62		
VSI	LU SEND Issued	Page TRC-60		
VSR	LU SEND Response	Page TRC-61		

CC - This field denotes the Change Control activity for the HNAS VRM level where changes were introduced or modified. The cc/vrm value identifiers are typically assigned for changes affecting commands, modifiers, formats or operational characteristics under the following circumstances: A blank or (N) CC value denotes that the enhancement was introduced in the initial release of the denoted vrm distribution level. A value of (A) Added, (C) Changed, (D) Deleted or (R) Retired denotes change control activity for an active vrm release.

VRM - This field refers to the HNAS Version, Release, Modification level for the change control implementation type.

CPU Trace Table Entries

(0) CPU Utilization (TRIDCPU for PARM=TRCTASK)

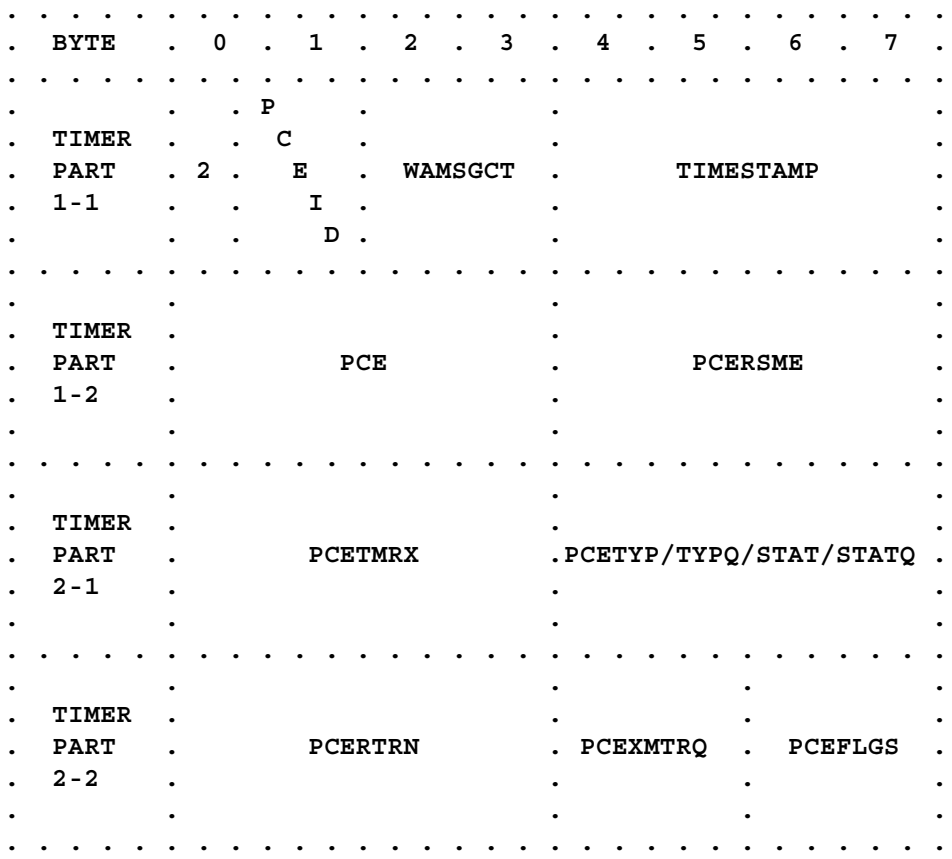
BYTE	0	1	2	3	4	5	6	7
CPU PART 1-1	0	DELTA TOD UTILIZATION						
CPU PART 1-2		TOTAL TOD UTILIZATION						

Dispatcher Trace Table Entries

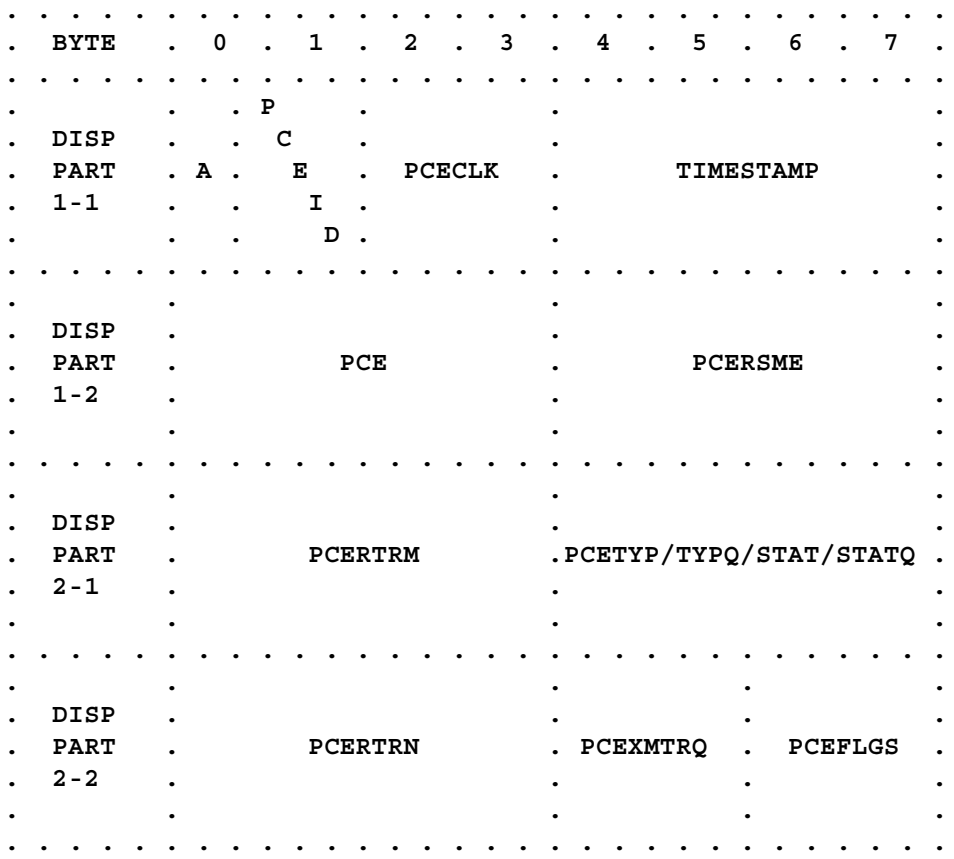
(1) Subtask Dispatch (TRIDDISP for PARM=TRCDISP)

BYTE	0	1	2	3	4	5	6	7
DISP PART 1-1	1	PCE	PCECLK		TIMESTAMP			
DISP PART 1-2	PCE				PCERSME			
DISP PART 2-1	PCETMRX				PCETYP/TYPQ/STAT/STATQ			
DISP PART 2-2	PCERTRN				PCEXMTRQ		PCEFLGS	

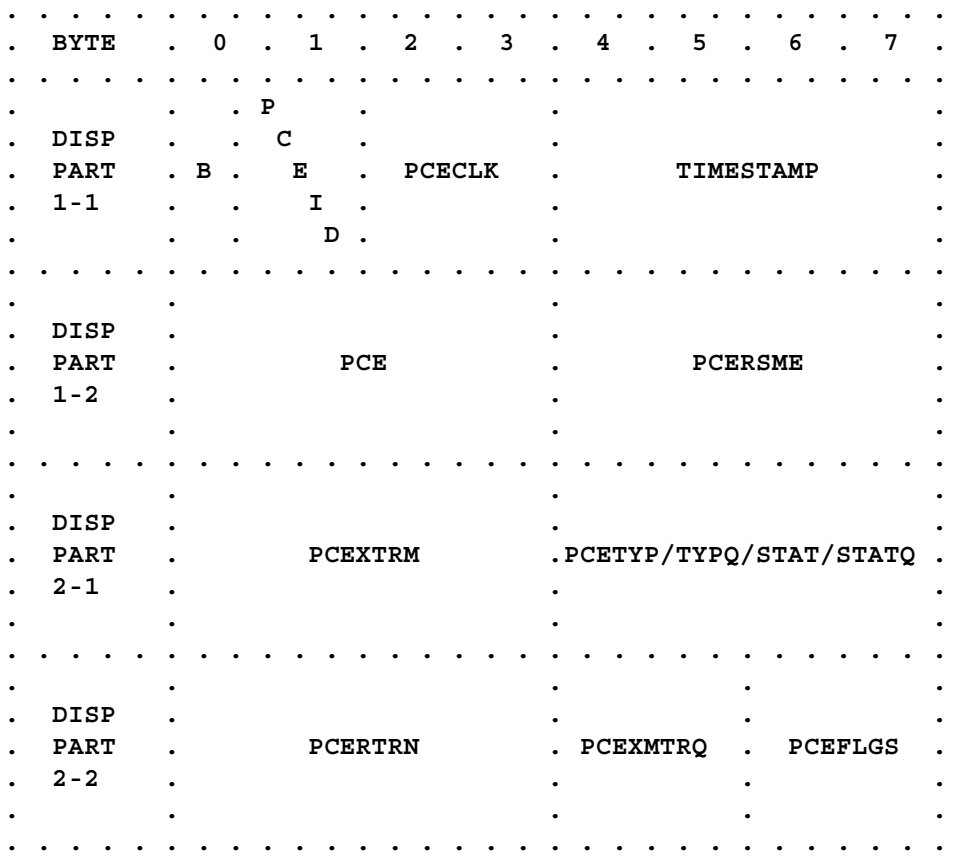
(2) Timer Completion (TRIDTMR for PARM=TRCDISP)



(A) TCP/IP Receive Completion (TRIDRTRM for PARM=TRCDISP)



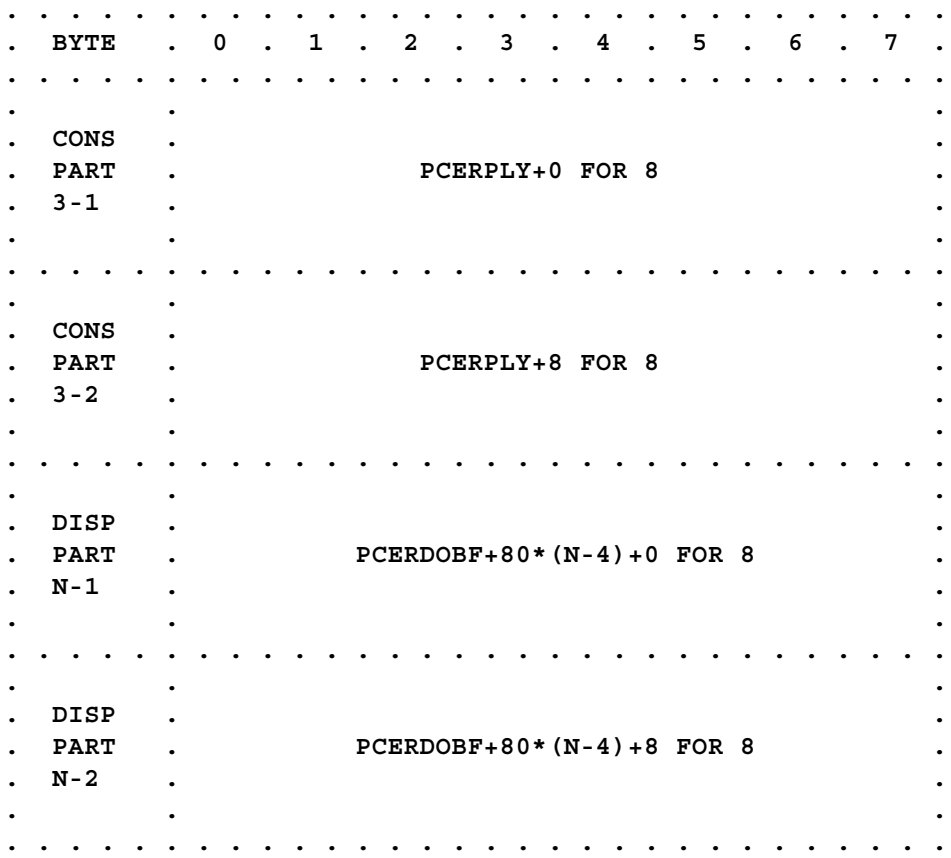
(B) TCP/IP Transmit Completion (TRIDXTRM for PARM=TRCDISP)



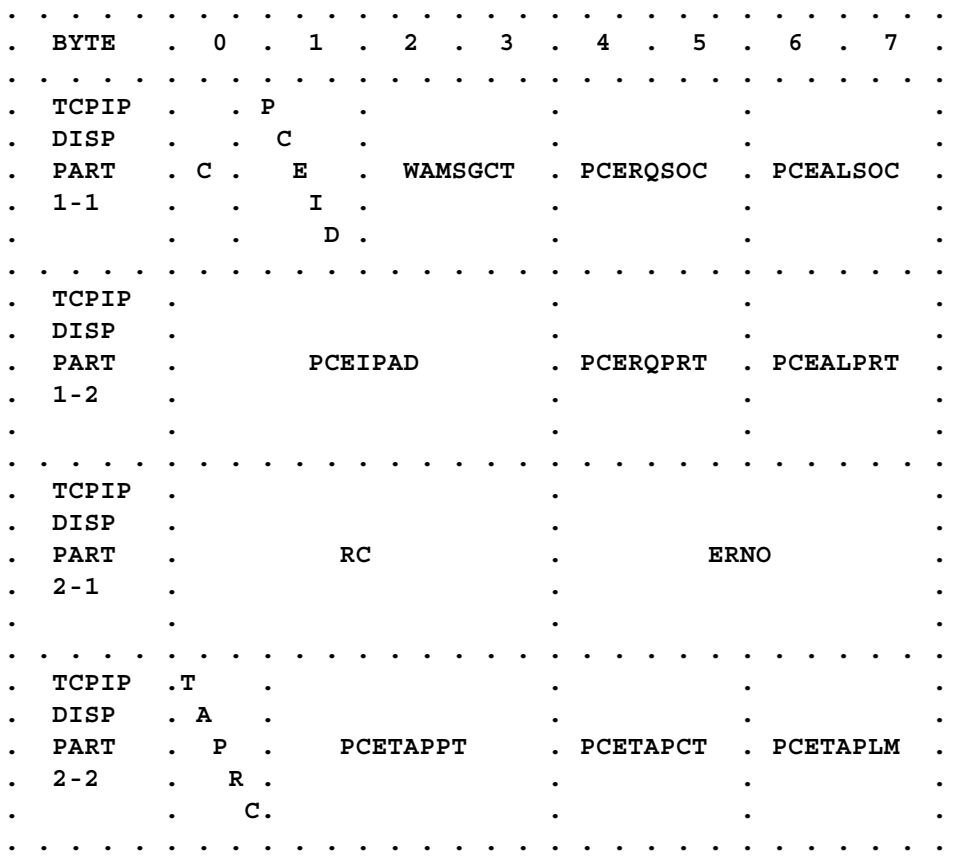
(9) Console Subtask Request (TRIDCONS for PARM=TRCDISP)

BYTE	0	1	2	3	4	5	6	7
CONS PART 1-1	9	PCEI	PCECLK		TIMESTAMP			
CONS PART 1-2	PCE				PCERSME			
CONS PART 2-1	PCETMRX				PCETYP/TYPQ/STAT/STATQ			
CONS PART 2-2	PCERTRN				PCERDOCT		PCERDOLM	

(9) Console Subtask Request (Continued)



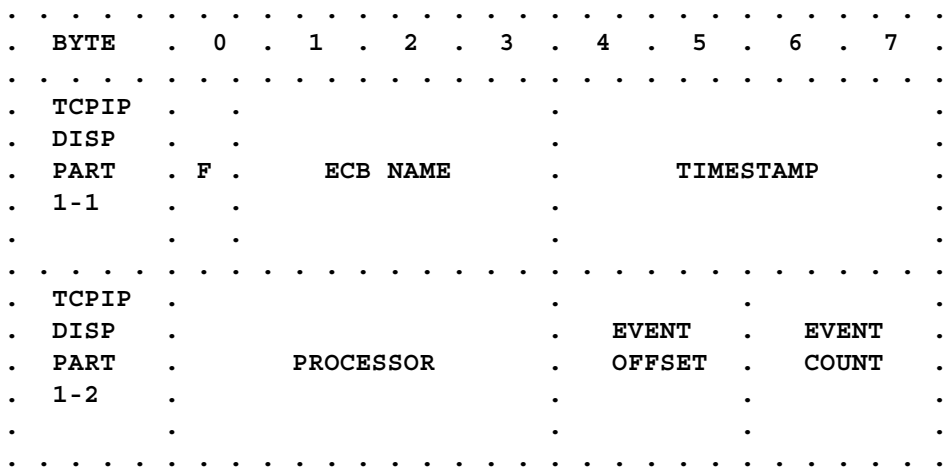
(C) TCP/IP Subtask Completion (TRIDTCP for PARM=TRCDISP)



(E) Main Task WAIT (TRIDWAIT for PARM=TRCTASK)

BYTE	0	1	2	3	4	5	6	7
TCPIP								
DISP								
PART	E					TIMESTAMP		
1-1								
TCPIP								
DISP		TOTAL				DELTA		
PART		EVENT				EVENT		
1-2		COUNT				COUNT		

(F) Main Task POST (TRIDPOST for PARM=TRCTASK)



TCP/IP I/O Trace Table Entries

(3) TCP/IP Request Parameter List (TRIDIPRM for PARM=TRCIO)

BYTE	0	1	2	3	4	5	6	7
IPRM PART 1-1	3	PCE I D	WAMSGCT		TIMESTAMP			
IPRM PART 1-2	PCE				PCERSME			
IPRM PART 2-1	PCERTRN				PCETGCLS		PCETGSOC	
IPRM PART 2-2	IPARM				IPPATHID		IPFLAGS1 AND IPRCODE	
					00		02	

(3) TCP/IP Request Parameter List (continued)

BYTE	0	1	2	3	4	5	6	7
IPRM PART 3-1	IPMSGLIM (ACCEPT)		IPMSGID (REPLY)			IPTRGCLS		
IPRM PART 3-2	IPBFADR1 OR IPRMMSG1				IPBFLN1F OR IPRMMSG2			
IPRM PART 4-1	IPSRCCLS				IPMSGTAG			
IPRM PART 4-2	IPBFADR2				IPBFLN2F			

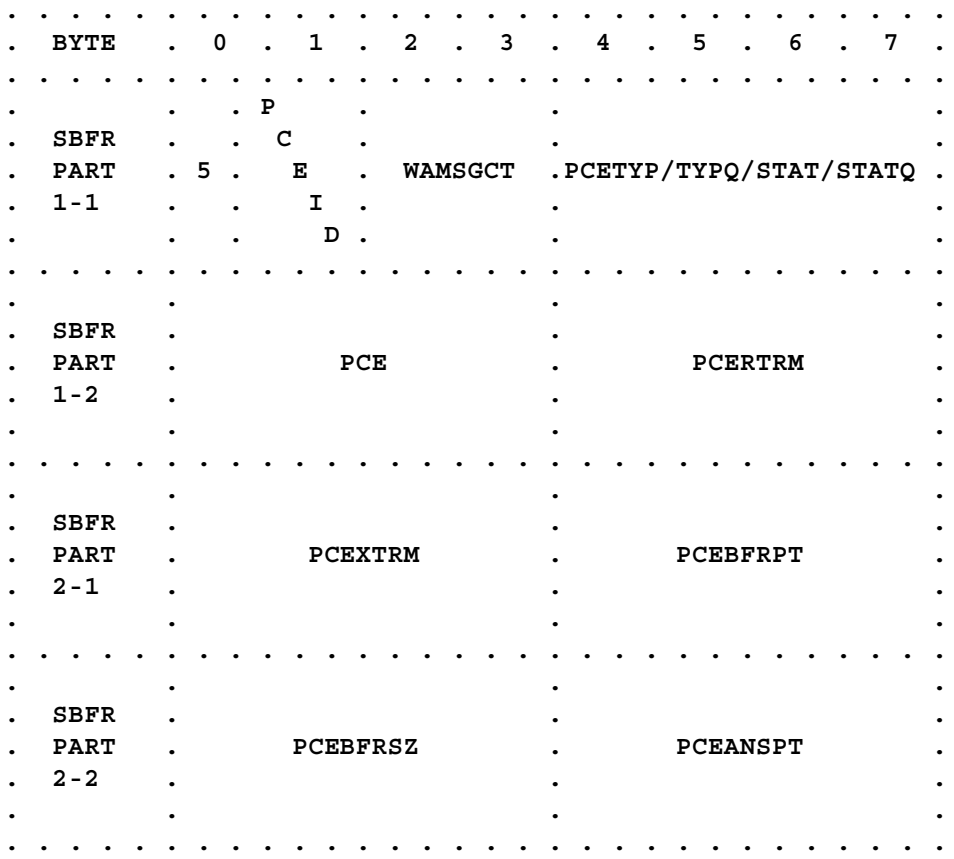
(4) TCP/IP External Interrupt Buffer (TRIDIBFR for PARM=TRCIO)

BYTE	0	1	2	3	4	5	6	7
IBFR PART 1-1	4	PCEI	WAMSGCT		TIMESTAMP			
IBFR PART 1-2	PCE				PCERSME			
IBFR PART 2-1	PCERTRN				PCETGCLS		PCETGSOC	
IBFR PART 2-2	BFR				IPPATHID 00		IPFLAGS1 AND IPTYPE 02	

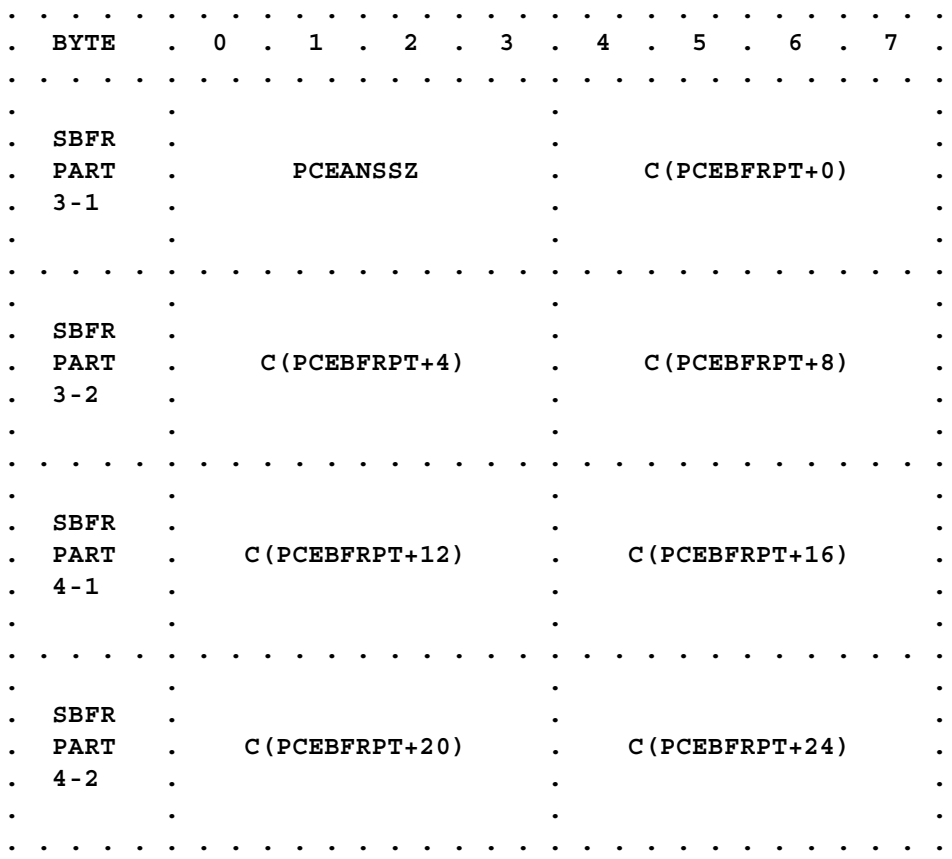
(4) TCP/IP External Interrupt Buffer (continued)

BYTE	0	1	2	3	4	5	6	7
IBFR PART 3-1	IPMSGLIM (ACCEPT)		IPMSGID (REPLY)		IPAUDIT			
	04				08			
IBFR PART 3-2	IPRMMSG1				IPRMMSG2			
	0C				10			
IBFR PART 4-1	IPSRCCLS				IPMSGTAG			
	14				18			
IBFR PART 4-2					IPBFLN2F			
	1C				20			

(5) TCP/IP Request Buffer (TRIDSBFR for PARM=TRCBFR)



(5) TCP/IP Request Buffer (continued)



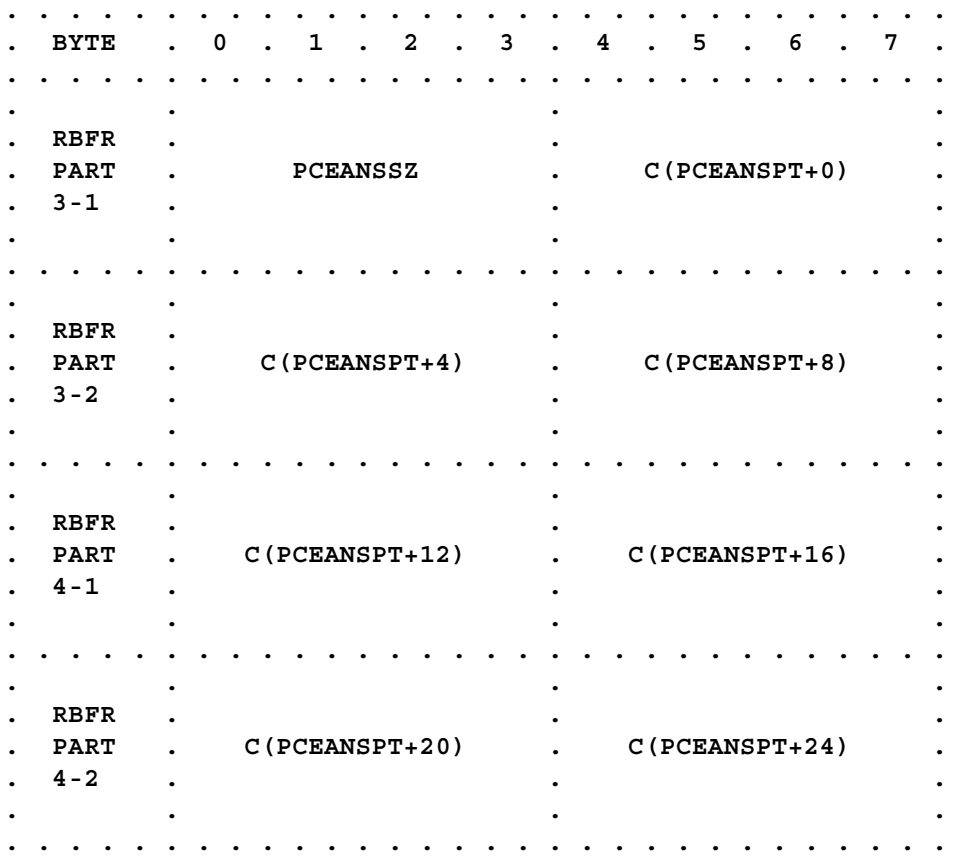
(5) TCP/IP Request Buffer (continued)

BYTE	0	1	2	3	4	5	6	7
RCV SBFR PART 5-1	PCERPKT				PCERBFR			
RCV SBFR PART 5-2	PCERBCT				PCERBSZ			
XMT SBFR PART 5-1	PCEXPKT				PCEXBFR			
XMT SBFR PART 5-2	PCEXRQ+0				PCEXRQ+4			

(6) TCP/IP Response Buffer (TRIDRBFR for PARM=TRCBFR)

BYTE	0	1	2	3	4	5	6	7
RBFR PART 1-1	6	P C E I D	WAMSGCT		PCETYP/TYPQ/STAT/STATQ			
RBFR PART 1-2	PCE				PCERTRM			
RBFR PART 2-1	PCEXTRM				PCEBFRPT			
RBFR PART 2-2	PCEBFRSZ				PCEANSPT			

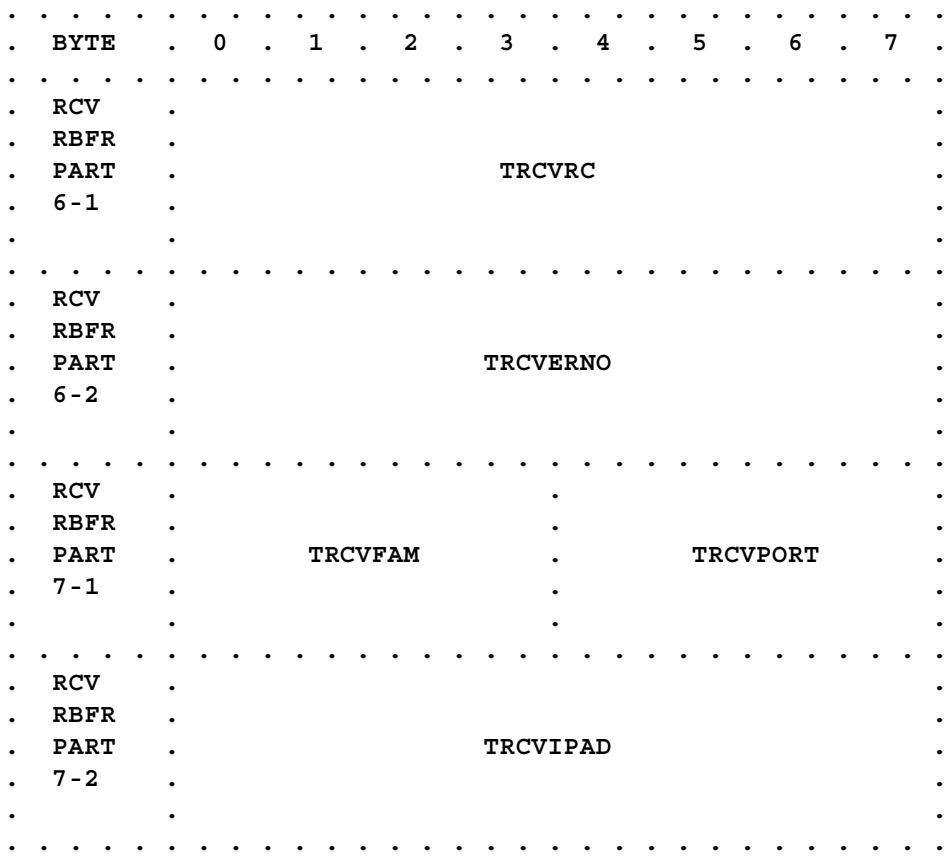
(6) TCP/IP Response Buffer (Continued)



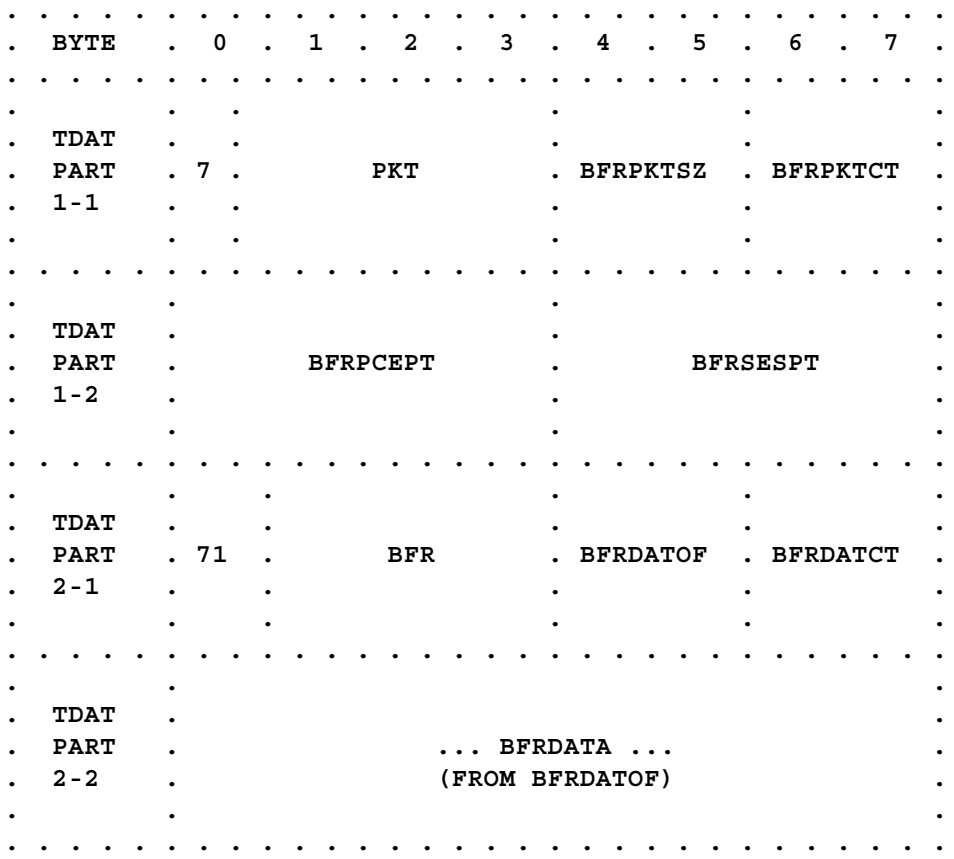
(6) TCP/IP Response Buffer (Continued)

BYTE	0	1	2	3	4	5	6	7
RCV RBFR PART 5-1		PCERPKT				PCERBFR		
RCV RBFR PART 5-2		PCERBCT				PCERBSZ		
XMT RBFR PART 5-1		PCEXPKT				PCEXBFR		
XMT RBFR PART 5-2		PCEXRQ+0				PCEXRQ+4		

(6) TCP/IP Response Buffer (Continued)

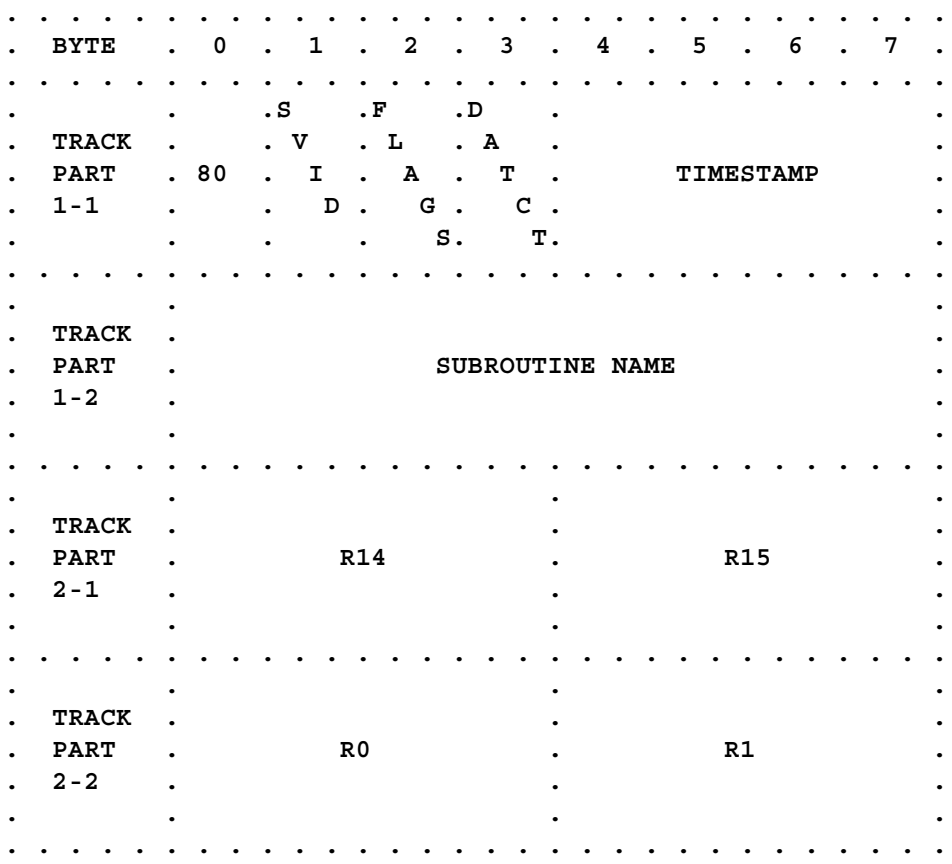


(7) TCP/IP Data Buffer (TRIDTDAT for PARM=TRCDATA)



Subroutine Call Trace Table Entries

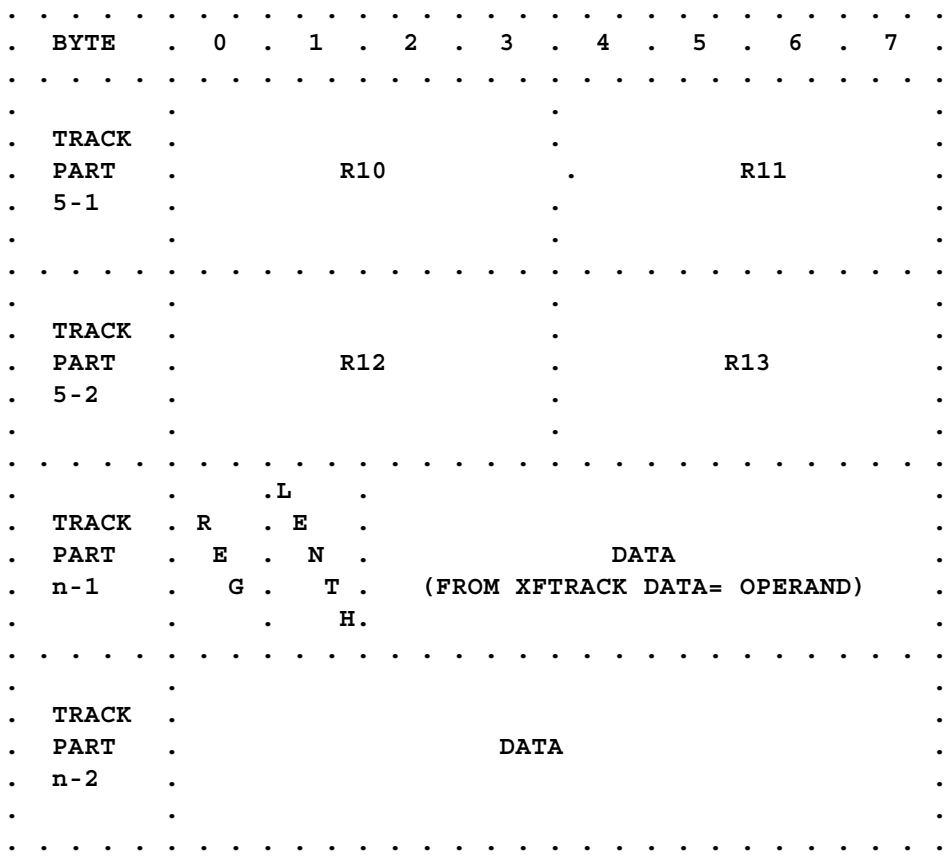
(80) Subroutine Call (TRIDSUBR for PARM=TRCSUBR)



(80) Subroutine Call (Continued)

BYTE	0	1	2	3	4	5	6	7
TRACK PART 3-1		R2				R3		
TRACK PART 3-2		R4				R5		
TRACK PART 4-1		R6				R7		
TRACK PART 4-2		R8				R9		

(80) Subroutine Call (Continued)

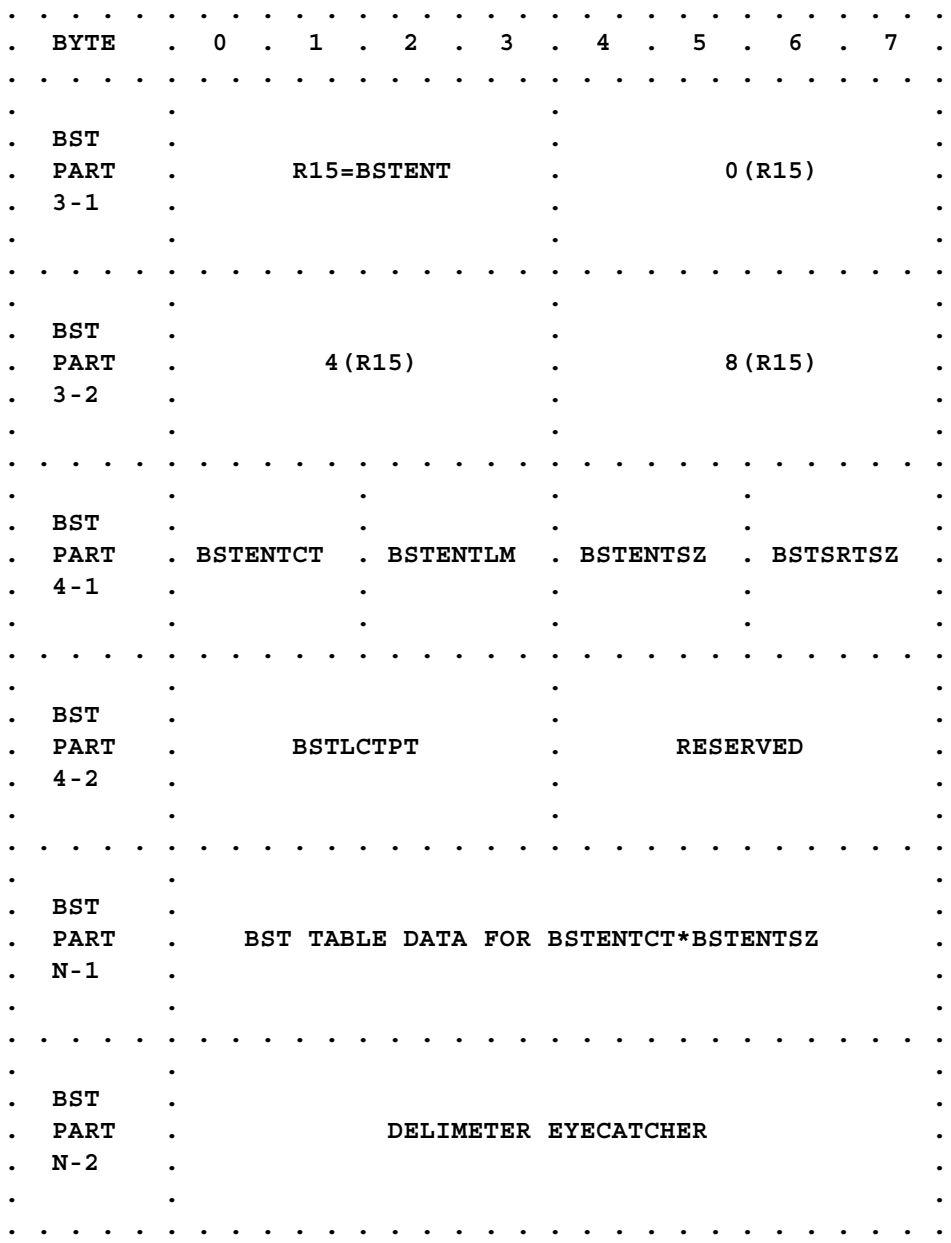


Subroutine Call Trace Table Entries

(81) Binary Search Call (TRIDBST for PARM=TRCBST)

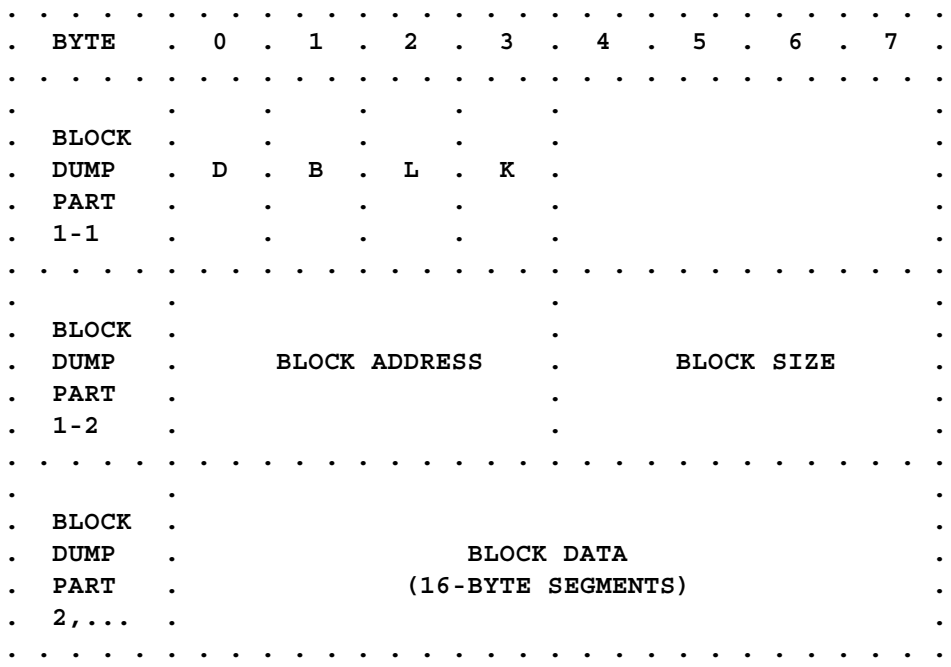
BYTE	0	1	2	3	4	5	6	7
BST PART 1-1	81	R0=BST			TIMESTAMP			
BST PART 1-2	CALLER EYECATCHER							
BST PART 2-1	R1=BSTARG				0(R1)			
BST PART 2-2	4(R1)				8(R1)			

(81) Binary Search Call (Continued)

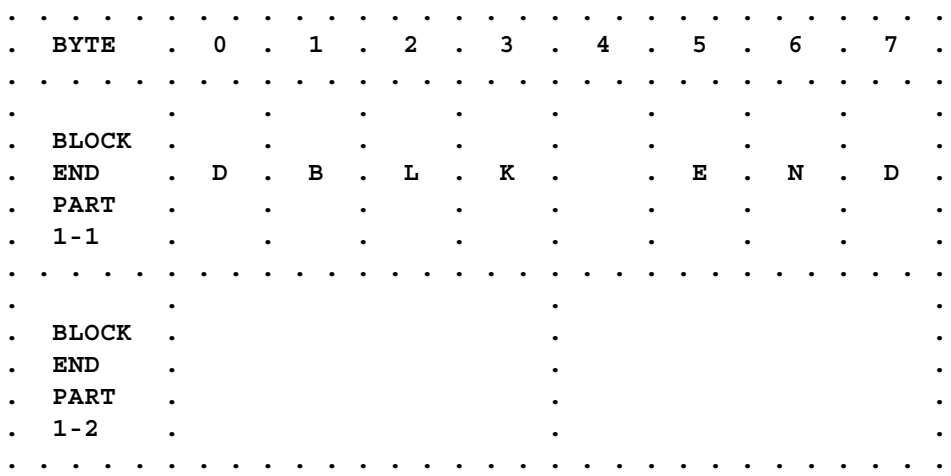


Common Component Trace Table Entries

(DBLK) Block Dump Start (for PARM=TRCDBK)



(DBLKEND) Block Dump End (for PARM=TRCDBK)



(GDIA) Gate Diagnostic Message (for PARM=TRCMCH)

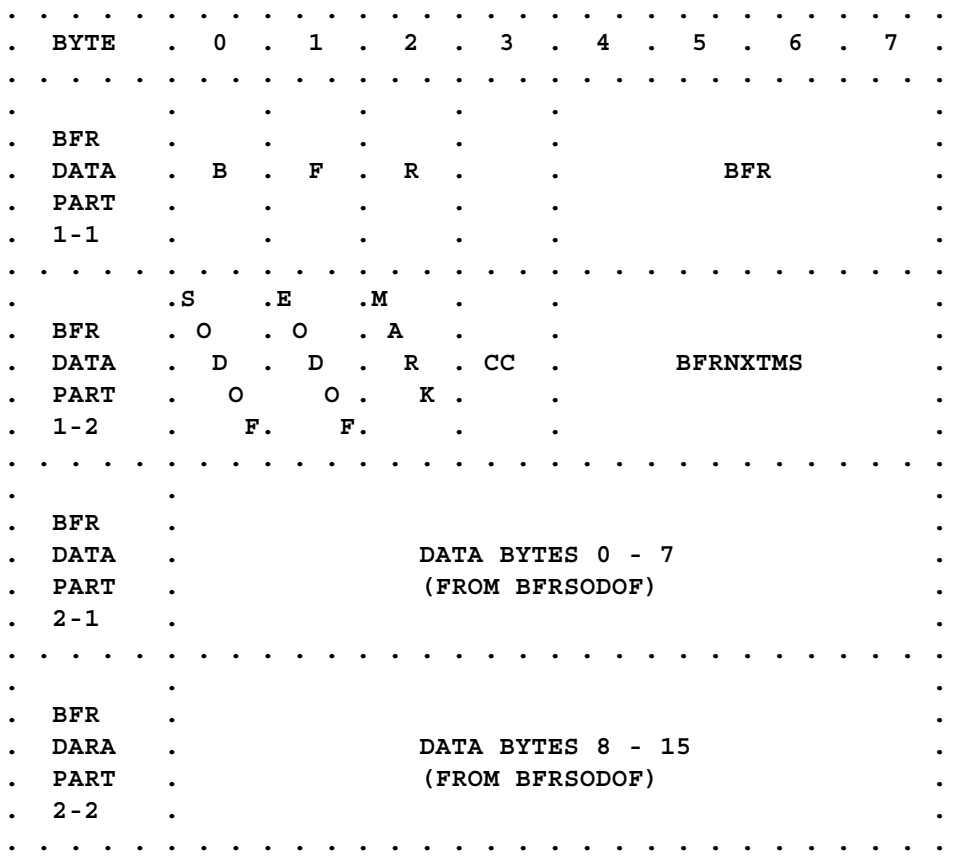
BYTE	0	1	2	3	4	5	6	7
DIAG MSG PART 1-1	G	D	I	A	TIMESTAMP			
DIAG MSG PART 1-2	MCH				LUB			
DIAG MSG PART 2-1	DATA BYTES 0 - 7							
DIAG MSG PART 2-2	DATA BYTES 8 - 15							

(XBt) Transmit Buffer Request (for PARM=TRCMCH)

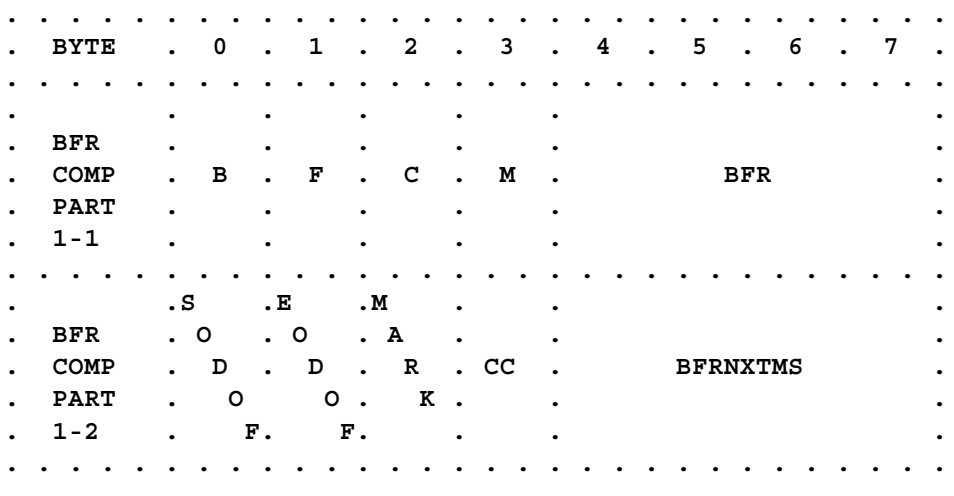
BYTE	0	1	2	3	4	5	6	7
XMT BFR PART 1-1	X	B	t		TIMESTAMP			
XMT BFR PART 1-2	MCH				VCB			
XMT BFR PART 2-1	LUB				BFR			
XMT BFR PART 2-2	MCHLXHCT	VCLXHCT		VC PACE	VCL XPKTR	LUB ST1	LUB ST2	

Legend: XBt
|
|<-- A => Allocate
F => Free

(BFR) Buffer Data (for PARM=TRCDATA)



(BFCM) Buffer Completion (for PARM=TRCDATA)



Virtual Circuit Trace Table Entries

(LSRS) VC Reset Request Scheduled (for PARM=TRCVC)

BYTE	0	1	2	3	4	5	6	7
RESET REQ SCHED PART 1-1	L	S	R	S	TIMESTAMP			
RESET REQ SCHED PART 1-2	MCH				VCB			
RESET REQ SCHED PART 2-1	LUB				VCRESET CALLER (R14)			
RESET REQ SCHED PART 2-2	VC LCST		CODE		LUB ST1	LUB ST2	LUIQ QCBCT	

(LSCL) VC Clear Request Scheduled (for PARM=TRCVC)

BYTE	0	1	2	3	4	5	6	7
CLEAR REQ SCHED PART 1-1	L	S	C	L	TIMESTAMP			
CLEAR REQ SCHED PART 1-2	MCH				VCB			
CLEAR REQ SCHED PART 2-1	LUB				VCCLEAR CALLER (R14)			
CLEAR REQ SCHED PART 2-2	VC LCST		CODE		LUB ST1	LUB ST2	LUIQ QCBCT	

(VCRL) VC Releasing Resources (for PARM=TRCVC)

BYTE	0	1	2	3	4	5	6	7
VC FREE PART 1-1	V	C	R	L	TIMESTAMP			
VC FREE PART 1-2	MCH				VCB			
VC FREE PART 2-1	LUB				VCVCRL CALLER (R14)			
VC FREE PART 2-2					LUB ST1	LUB ST2	LUIQ QCBCT	

(VCTO) VC Timeout (for PARM=TRCVC)

BYTE	0	1	2	3	4	5	6	7
VC TMOUT PART 1-1	V	C	T	O	TIMESTAMP			
VC TMOUT PART 1-2	MCH				VCB			
VC TMOUT PART 2-1	LUB				VC LCST	VC CLOCK		
VC TMOUT PART 2-2					LUB ST1	LUB ST2	LUIQ QCBCT	

(RTMC) XTP MCH Receive Data (for PARM=TRCMCH)

BYTE	0	1	2	3	4	5	6	7
XTP MCH RCV PART 1-1	R	T	M	C	TIMESTAMP			
XTP MCH RCV PART 1-2	MCH							

(RTCR) XTP VC Receive Call Request (for PARM=TRCMCH)

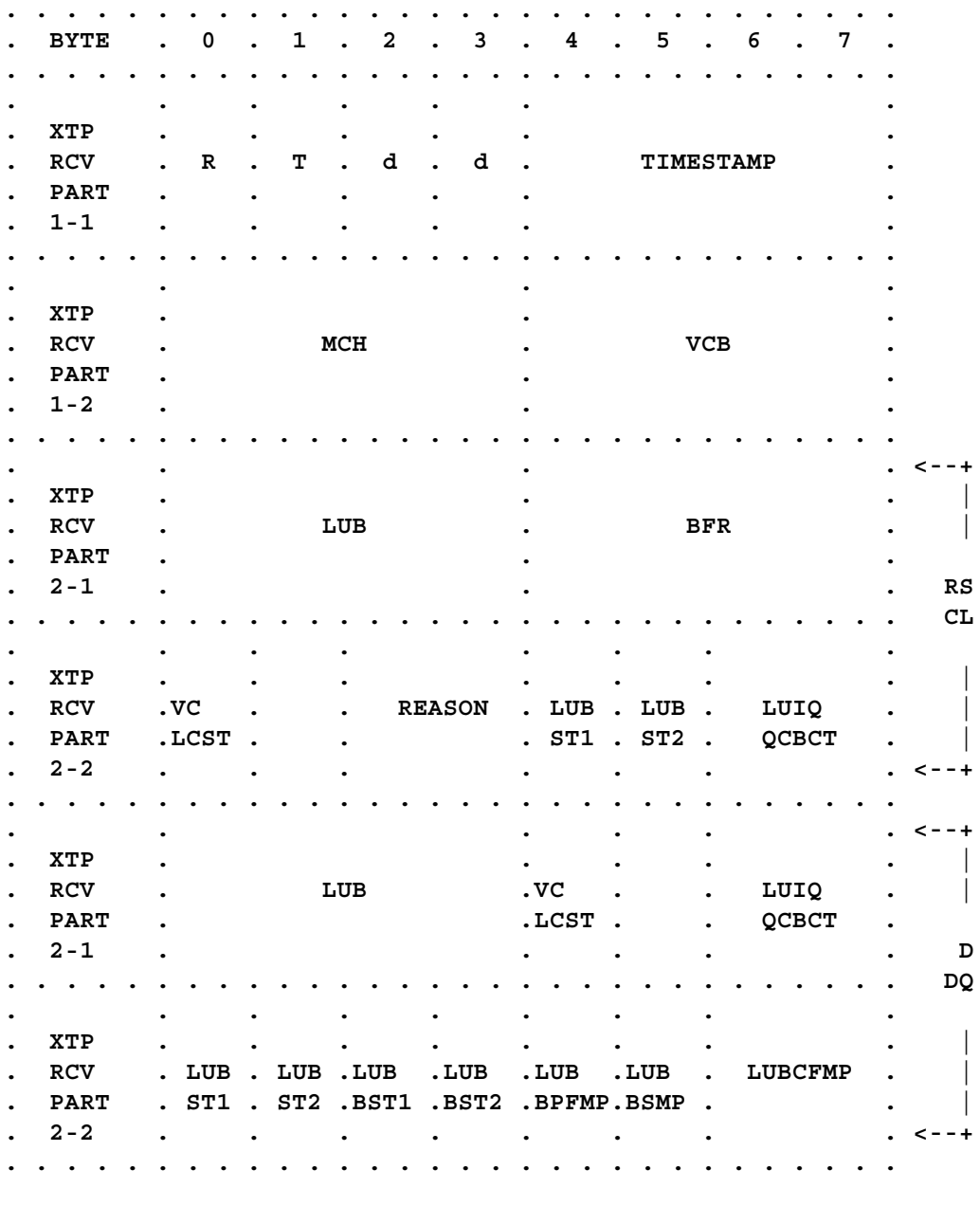
BYTE	0	1	2	3	4	5	6	7
XTP RCV CLRQ PART 1-1	R	T	C	R	TIMESTAMP			
XTP RCV CLRQ PART 1-2	MCH							
XTP RCV CLRQ PART 2-1	M	B	F	R	BFR			
XTP RCV CLRQ PART 2-2								

(RTCRFAIL) XTP Receive Call Request Failure (for PARM=TRCMCH)

BYTE	0	1	2	3	4	5	6	7
XTP RCV CLRQFL PART 1-1	R	T	C	R	F	A	I	L
XTP RCV CLRQFL PART 1-2					C A U S E.	D I A G		
XTP RCV CLRQFL PART 2-1	M	B	F	R		BFR		
XTP RCV CLRQFL PART 2-2								

MCH

(RTdd) XTP VC Receive Data (for PARM=TRCMCH)



Legend: RTdd (see next page)

```
Legend: RTdd
          |
          |<-- ** => Invalid Type
               D  => Data
               DQ => Qualified Data
               RR => Receiver Ready
               RN => Receiver Not Ready
               RS => Reset Request
               CL => Clear Request
               CR => Call Request
               CA => Call Accept
               PI => PVC Information
               PD => PVC Down
```

Notes:

1) A single 16-byte trace entry (Part 1-1 and 1-2) is logged for RR, RN, CR, CA, PI and PD trace requests.

2) Two 16-byte trace entries are logged for RS, CL, D and DQ trace requests. Differences in the data logged for Part 2-1 and 2-2 is identified in the chart above.

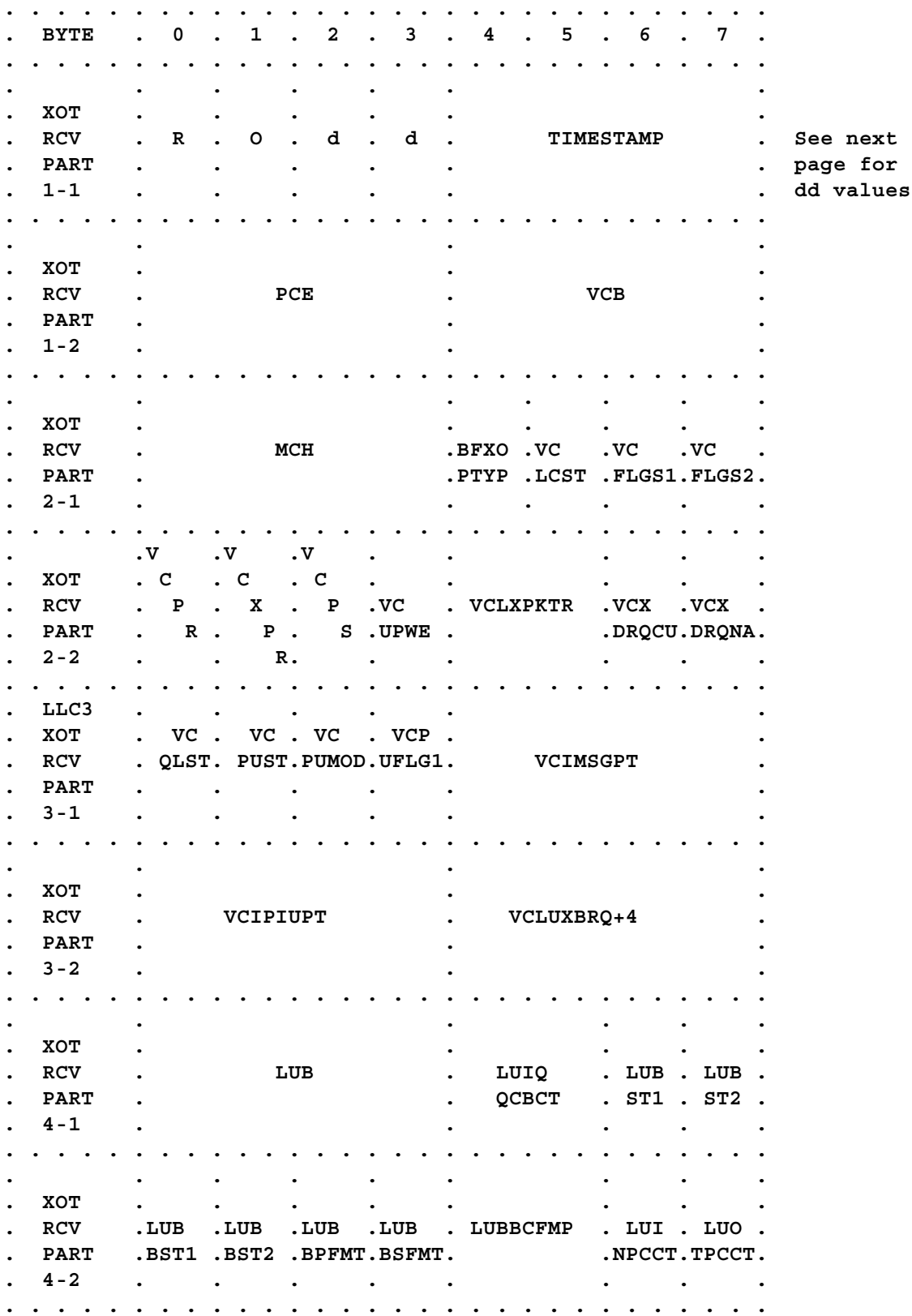
(ROCR) XOT VC Receive Call Request (for PARM=TRCMCH)

BYTE	0	1	2	3	4	5	6	7
XOT RCV CLRQ PART 1-1	R	O	C	R	TIMESTAMP			
XOT RCV CLRQ PART 1-2	IPADDR				PORT		MCH	
XOT RCV CLRQ PART 2-1	M	B	F	R	BFR			
XOT RCV CLRQ PART 2-2								

(ROCRFAIL) XOT Receive Call Request Failure (for PARM=TRCMCH)

BYTE	0	1	2	3	4	5	6	7
XTP								
RCV								
CLRQFL	R	O	C	R	F	A	I	L
PART								
1-1								
XTP					C	D		
RCV					A	I		
CLRQFL	IPADDR				U	A		
PART					S	G		
1-2					E.			
XTP								
RCV								
CLRQFL	M	B	F	R	BFR			
PART								
2-1								
XTP								
RCV								
CLRQFL								
PART								
2-2								

(ROdd) XOT VC Receive Data (for PARM=TRCMCH)



```
Legend: ROdd
          |
          |<-- ** => Invalid Type
               D  => Data
               DQ => Qualified Data
               RR => Receiver Ready
               RN => Receiver Not Ready
               RJ => Reject
               I  => Interrupt Request
               IC => Interrupt Confirm
               RS => Reset Request
               RC => Reset Confirm
               CL => Clear Request
               CC => Clear Confirm
               CR => Call Request
               CA => Call Accept
               DI => Diagnostic
               PS => PVC Setup
```

(Xpdd) VC Transmit Data (for PARM=TRCVC)

BYTE	0	1	2	3	4	5	6	7
VC XMT PART 1-1	X	p	d	d	TIMESTAMP			
VC XMT PART 1-2	MCH				VCB			
VC XMT PART 2-1	LUB				BFR			
VC XMT PART 2-2	VC LCST	VC PACE	VCLXHCT		LUB ST1	LUB ST2	LUIQ QCBCT	

Legend: Xpdd (see next page)

```
Legend: Xpdd
          ||
          ||<-- ** => Invalid Type
          |     D  => Data
          |     DQ => Qualified Data
          |     RR => Receiver Ready
          |     RN => Receiver Not Ready
          |     RJ => Reject (XOT only)
          |     I  => Interrupt Request (XOT only)
          |     IC => Interrupt Confirm (XOT only)
          |     RS => Reset Request
          |     RC => Reset Confirm (XOT only)
          |     CL => Clear Request
          |     CC => Clear Confirm (XOT only)
          |     CR => Call Request
          |     CA => Call Accept
          |     DI => Diagnostic (XOT only)
          |     PS => PVC Setup (XOT only)
          |     KA => Keep Alive (XTP only)
          |     PI => PVC Information (XTP only)
          |     PD => PVC Down (XTP only)
          |
          |<-- T => XTP
               O => XOT
```

(XpXC) VC Transmit Completion (for PARM=TRCVC)

BYTE	0	1	2	3	4	5	6	7
VC XMT COMP PART 1-1	X	p	X	C	TIMESTAMP			
VC XMT COMP PART 1-2	MCH|PCE (XTP|XOT)				BFR			
VC XMT COMP PART 2-1	BFRSESPT			CC	.X T A	.M R K P.		
VC XMT COMP PART 2-2	MCHLXHCT	VCLXHCT			VCB			

Legend: XpXC
|
|<-- T => XTP
O => XOT

Logical Unit Trace Table Entries

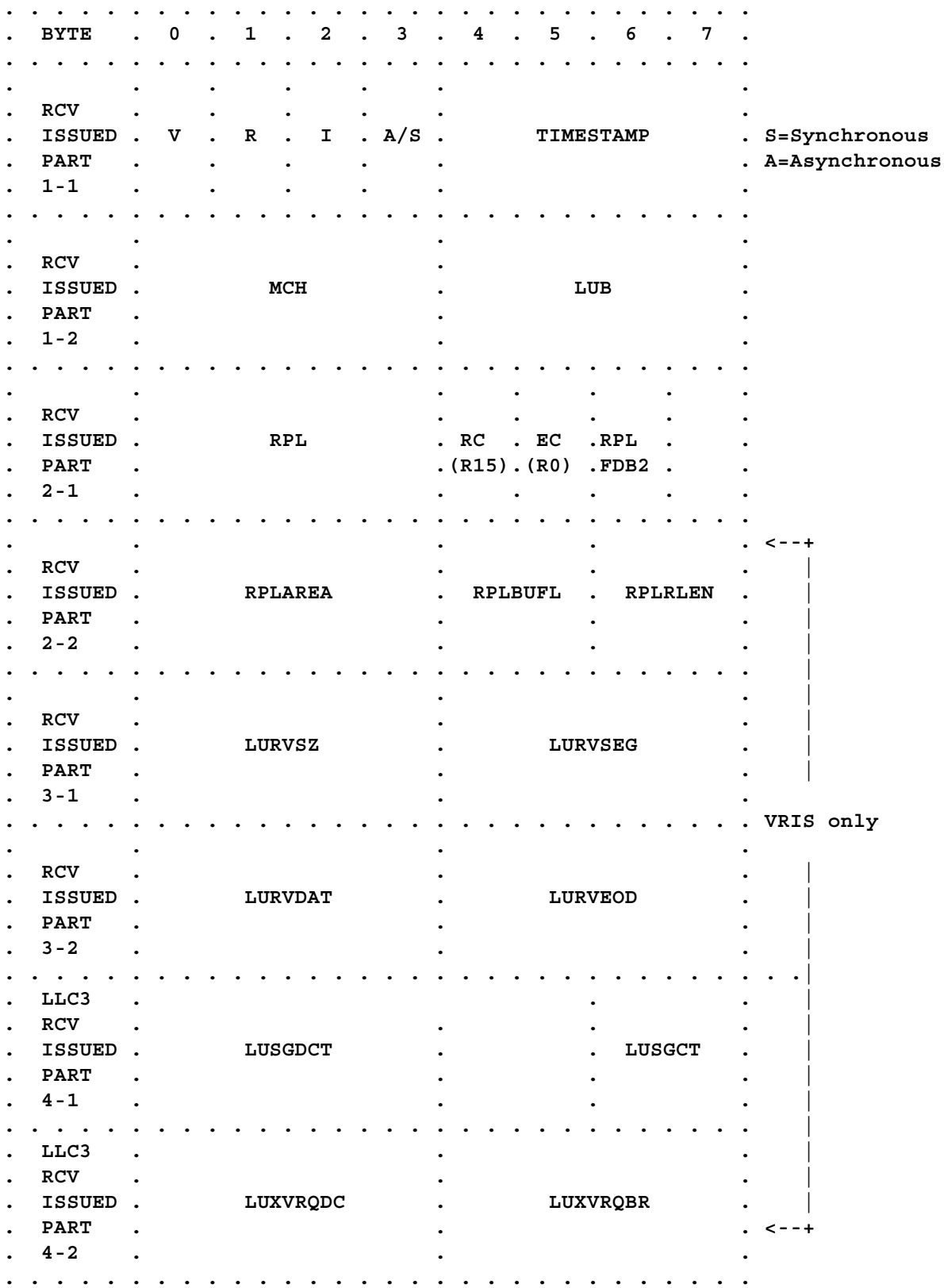
(VOPN) LU Open ACB (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
OPEN ACB PART 1-1	V	O	P	N	TIMESTAMP			
OPEN ACB PART 1-2	MCH				LUB			
OPEN ACB PART 2-1	RC (R15)	ACB ERFLG						
OPEN ACB PART 2-2	MCHX (LUMCHX)				VCB (LUVC)			

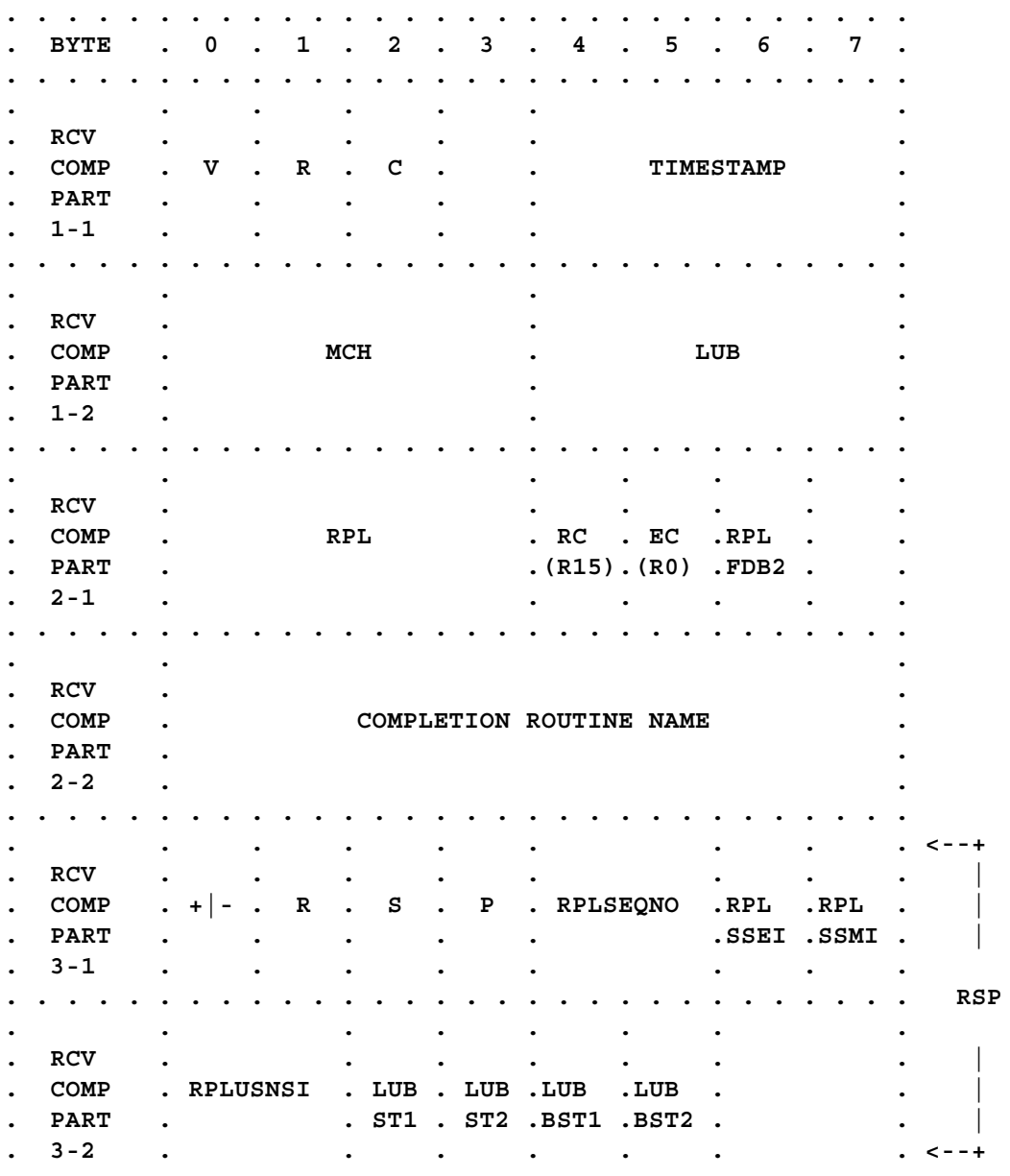
(VCLO) LU Close ACB (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
CLOSE ACB PART 1-1	V	C	L	O	TIMESTAMP			
CLOSE ACB PART 1-2	MCH				LUB			
CLOSE ACB PART 2-1	RC (R15)	ACB ERFLG						
CLOSE ACB PART 2-2	MCHX (LUMCHX)				VCB (LUVC)			

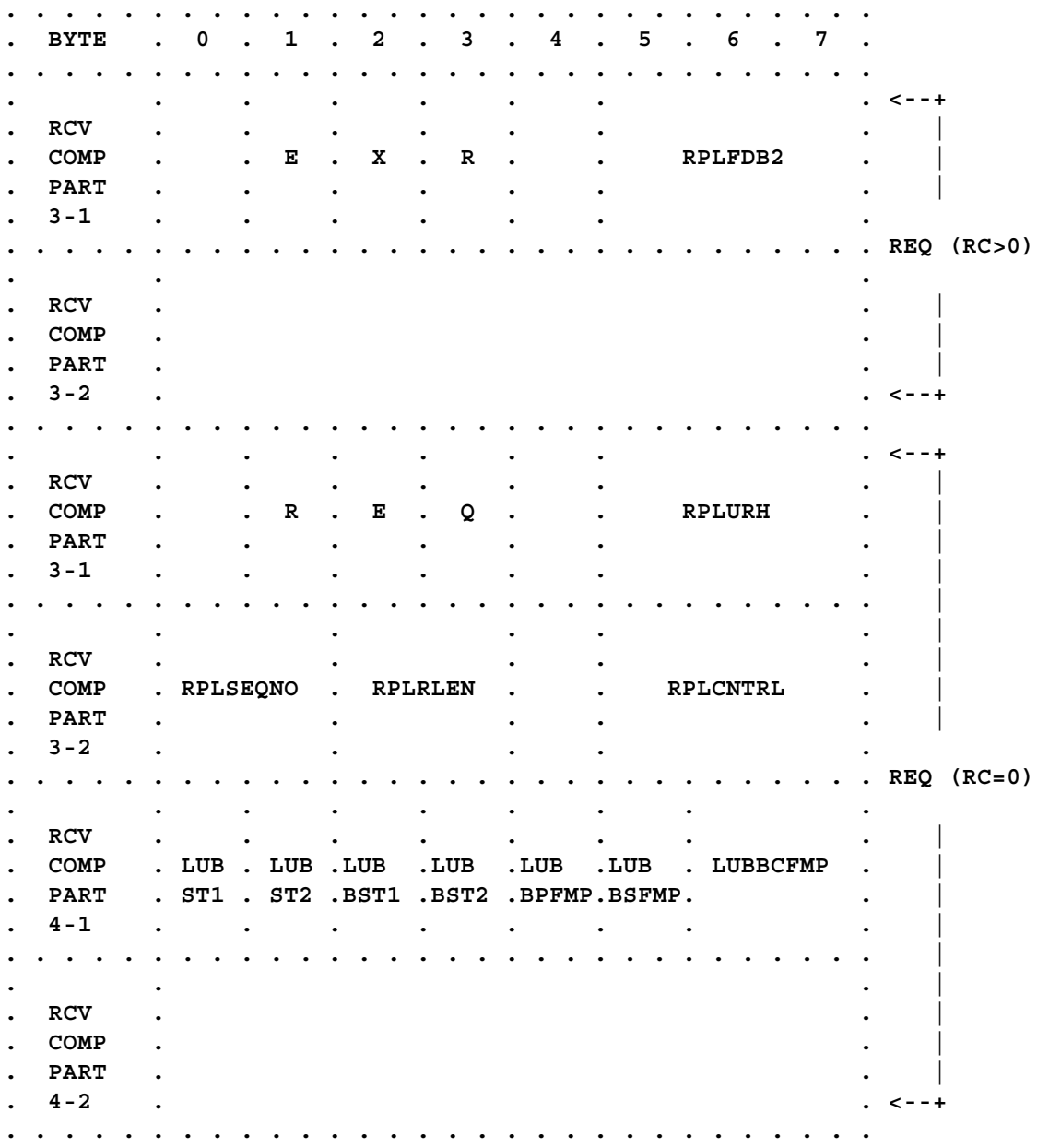
(VRIS / VRIA) LU Receive Issued (for PARM=TRCLU)



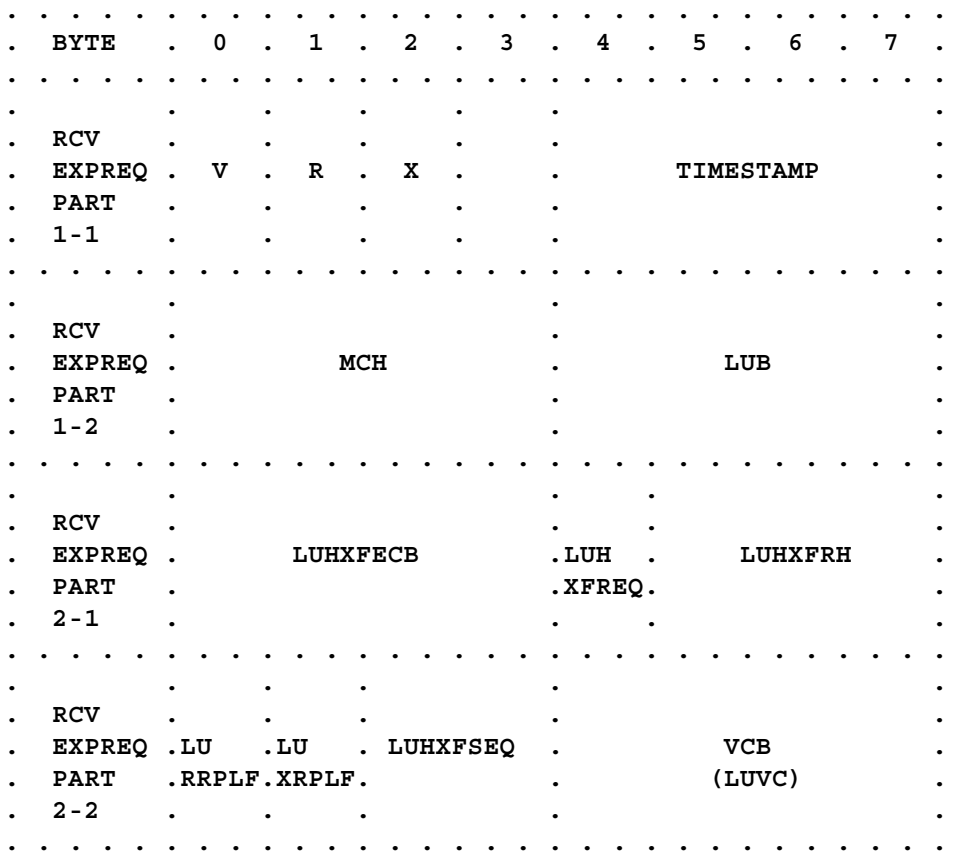
(VRC) LU Receive Completion (for PARM=TRCLU)



(VRC) LU Receive Completion (continued)



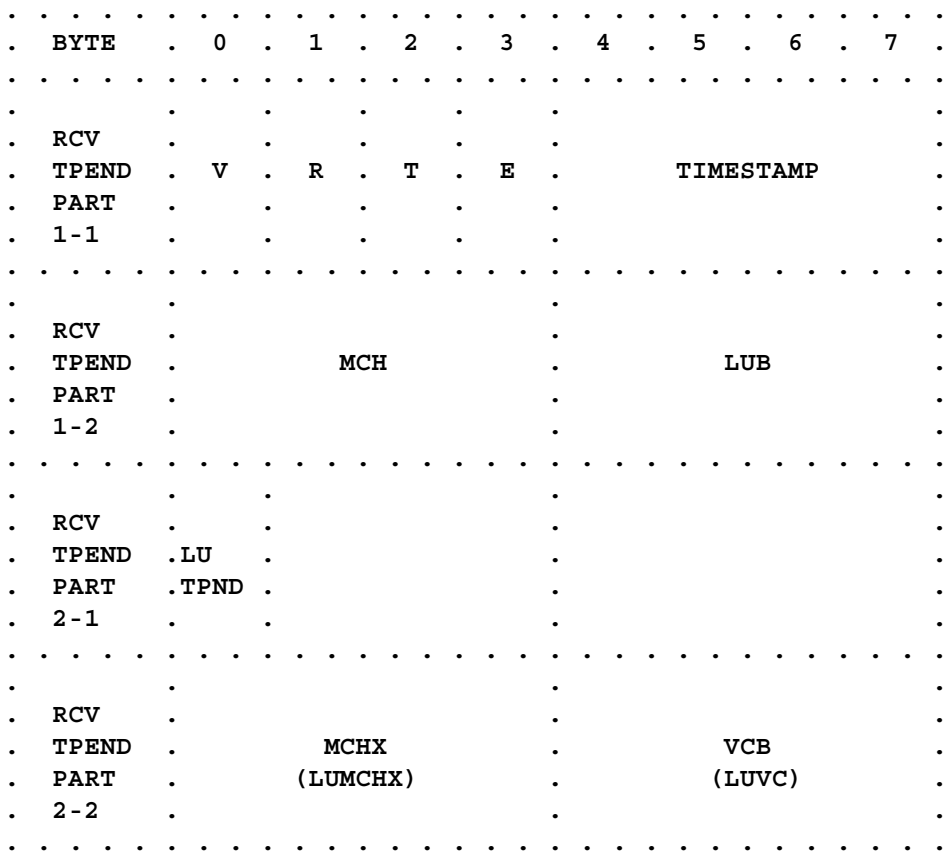
(VRX) LU Receive Expedited Request (for PARM=TRCLU)



(VRN) LU Receive Notify Request (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
RCV NOTIFY PART 1-1	V	R	N		TIMESTAMP			
RCV NOTIFY PART 1-2	MCH				LUB			
RCV NOTIFY PART 2-1	LUNTFY							
RCV NOTIFY PART 2-2	MCHX (LUMCHX)				VCB (LUVC)			

(VRTE) LU Receive TPEND Request (for PARM=TRCLU)



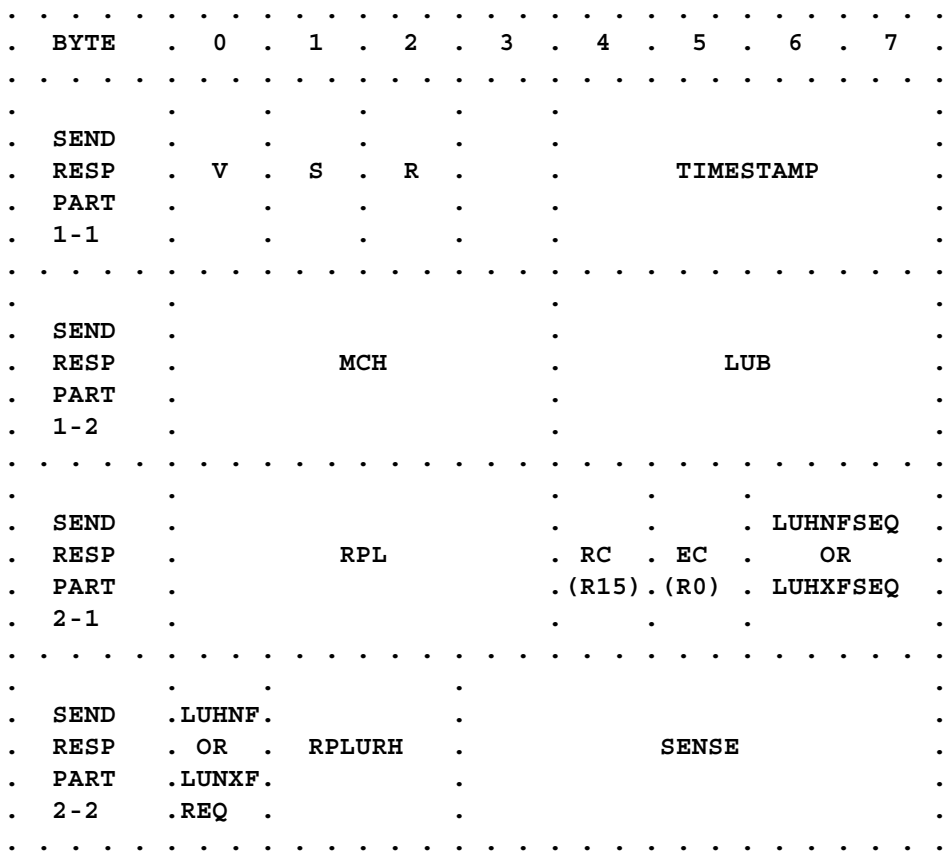
(VRF) LU Receive Flush (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
RCV FLUSH PART 1-1	V	R	F		TIMESTAMP			
RCV FLUSH PART 1-2	MCH				LUB			
RCV FLUSH PART 2-1	RPL				RC (R15)	EC (R0)		
RCV FLUSH PART 2-2	MCHX (LUMCHX)				VCB (LUVC)			

(VSI) LU Send Issued (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
SEND ISSUED PART 1-1	V	S	I		TIMESTAMP			
SEND ISSUED PART 1-2	MCH				LUB			
SEND ISSUED PART 2-1	RPL				RC (R15)	EC (R0)		
SEND ISSUED PART 2-2	REQUEST: CANCEL, DATA, LUSTAT, OPNSEC, REQSESS, RESETSR, RTR, SESSIONC, SETLOGON, SHUTC, SHUTD, UNBIND							

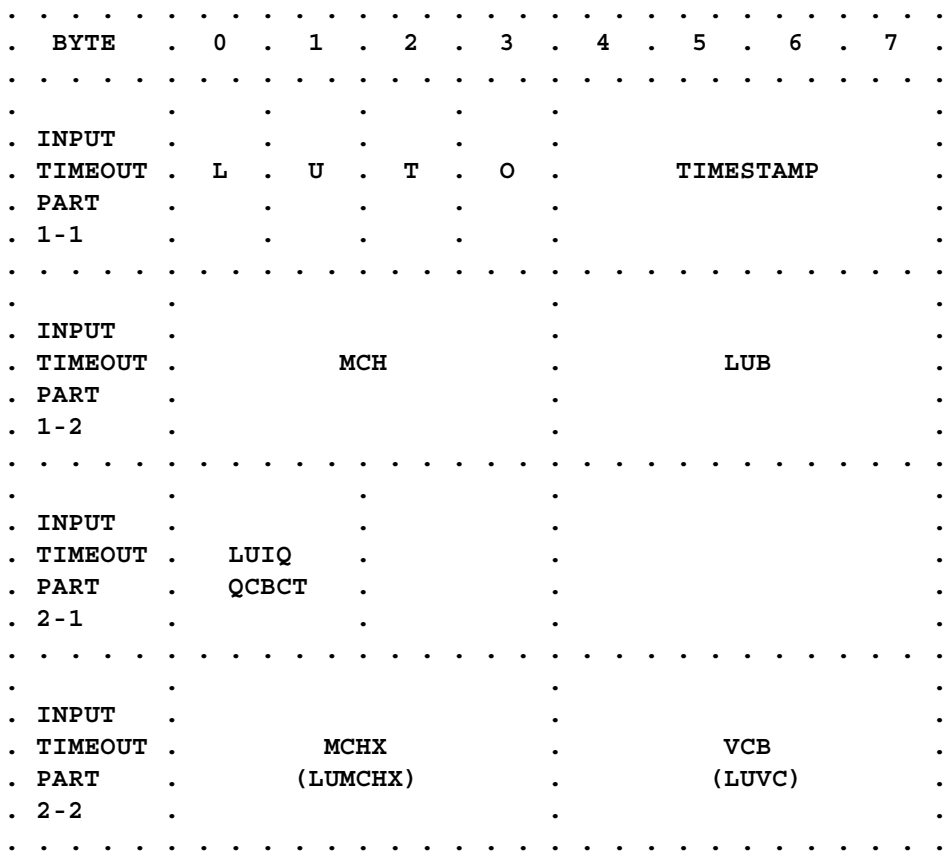
(VSR) LU Send Response (for PARM=TRCLU)



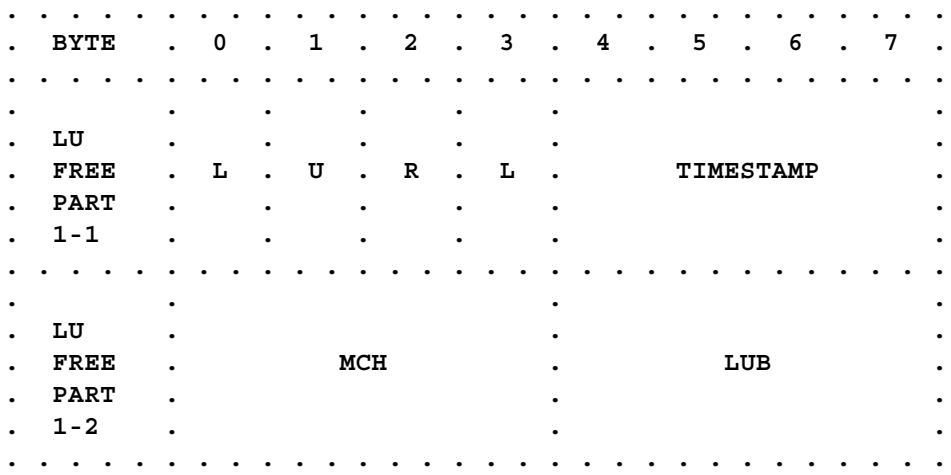
(VSC) LU Send Completion (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
SEND COMP PART 1-1	V	S	C		TIMESTAMP			
SEND COMP PART 1-2	MCH				LUB			
SEND COMP PART 2-1	RPL				RC (R15)	EC (R0)	RPL FDB2	
SEND COMP PART 2-2	COMPLETION ROUTINE NAME							

(LUTO) LU Input Queue Timeout (for PARM=TRCLUQ)



(LURL) LU Releasing Resources (for PARM=TRCLU)

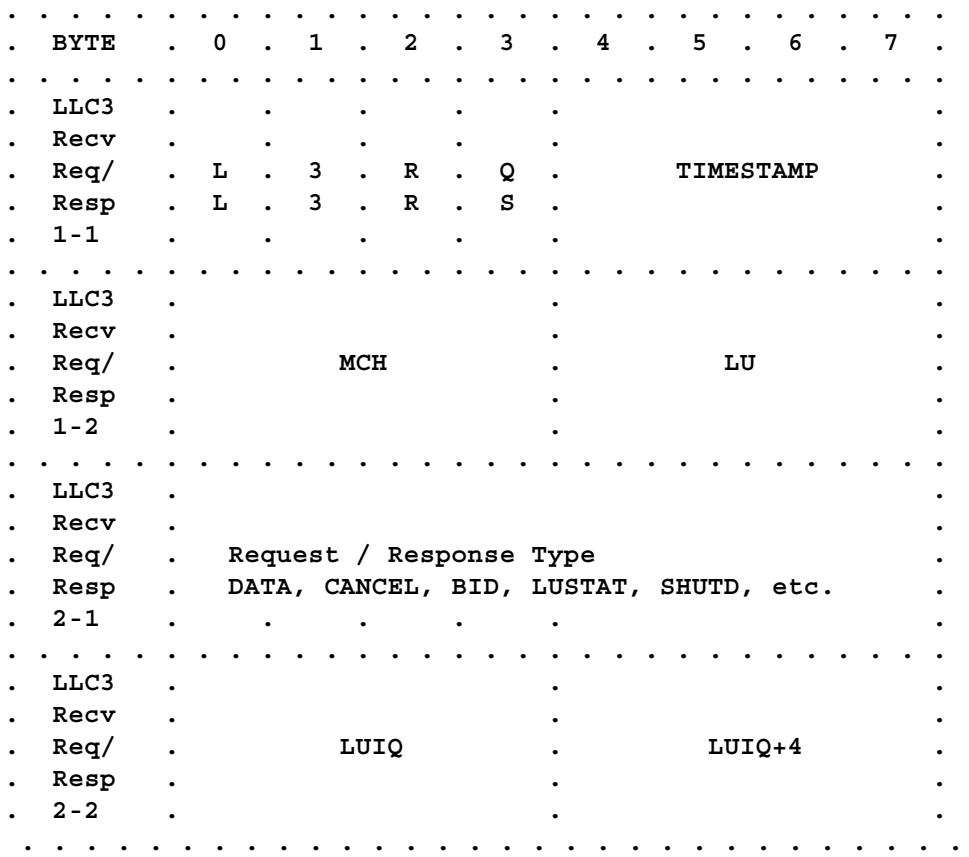


(LUSE) LU MCH Input State Error (for PARM=TRCLU)

BYTE	0	1	2	3	4	5	6	7
MCH STATE ERROR PART 1-1	L	U	S	E	TIMESTAMP			
MCH STATE ERROR PART 1-2	MCH				VCB			
MCH STATE ERROR PART 2-1	C O D E		LUBBCFMP		LUB ST1	LUB ST2	LUB BST1	LUB BST2
MCH STATE ERROR PART 2-2	LUIQ+0				LUIQ+4			

(L3RQ/L3RS) LU MCH LLC3 Receive REQ/RESP (for PARM=TRCLU)

(added in 220)



Multi-Channel Link SYSPRINT Trace Records

MCH SYSPRINT Trace Records

```
LU name LU addr ALOC BFR AT addr MCH XMT HDR CT=xxxx
LU name LU addr FREE BFR AT addr MCH XMT HDR CT=xxxx
LU name LU addr RMT name VC addr
        ALOC BFR AT xxxxxxxx MCH X-HDR=xxxx VC X-HDR=xxxx VCPACE=xx
LU name LU addr RMT name VC addr
        FREE BFR AT xxxxxxxx MCH X-HDR=xxxx VC X-HDR=xxxx VCPACE=xx
LU name LU addr
        SENDING DIAG PKT: xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx BFR NEXT
REMOTE MCH name BUFFER COMPRESSED
```

Virtual Circuit SYSPRINT Trace Records

VC SYSPRINT Trace Records

```
RMT name VC addr LU name LU addr RESET BY RTN name CODE=xxxx LCST=Pxx
RMT name VC addr LU name LU addr CLEARED BY RTN name CODE=xxxx LCST=Pxx

RMT name VC addr LU name LU addr XOT XMT TYPE=xx CALL RQ VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx CALL AC VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx DATA    VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx DATA-Q  VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx RR      VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx RNR     VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx REJ     VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx INT     VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx INT-C   VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx RESET   VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx RESET-C VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx CLEAR   VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx CLEAR-C VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx DIAG    VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XOT XMT TYPE=xx PVC SUP VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx CALL RQ VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx CALL AC VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx DATA    VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx DATA-Q  VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx RR      VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx RNR     VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx RESET   VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx CLEAR   VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx KEEP AL VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx PVC INF VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr XTP XMT TYPE=xx PVC DWN VC HDR CT=xxxx VCPACE=xx

RMT name VC addr LU name LU addr TIMEOUT CODE=xx LCST=Pxx VC HDR CT=xxxx VCPACE=xx
RMT name VC addr LU name LU addr VC REL W/ XMT HDR CT=xxxx
```

XTP Events SYSPRINT Trace Records

XTP SYSPRINT Trace Records

```
MCH addr RECV BUFFER NEXT
RMT name VC addr LU name LU addr XMIT COMP, BFR=xxxxxxxx TYP=xx CC=xx
MCH addr CALL REQ:
MCH addr PVC INFO:
MCH addr INBOUND CALL REQ FAILED, CLEAR DIAG=XX

RMT name VC addr LU name LU addr RCV  DATA LCST=Pxx BFR=xxxxxxxx
RMT name VC addr LU name LU addr RCV QDATA LCST=Pxx BFR=xxxxxxxx

LUIQCT=xxxx LUBST1/2=xx/xx LUBBST1/2=xx/xx LUBBPFMP/SFMP=xxxx
            LUBBCFMP=xxxx NF/XF REQ=xx/xx R/X RPLF=xx/xx

RMT name VC addr LU name LU addr  RCV RESET CODE=xxxx LCST=Pxx
RMT name VC addr LU name LU addr  RCV CLEAR CODE=xxxx LCST=Pxx
RMT name VC addr LU name LU addr  RCV RR
RMT name VC addr LU name LU addr  RCV RNR
RMT name VC addr LU name LU addr  RCV CALL ACCEPT
RMT name VC addr LU name LU addr  RCV PVC DOWN
```

XOT Events SYSPRINT Trace Records

XOT SYSPRINT TRACE RECORDS

```
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV CALL RQ LCST=PXX
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV CALL AC LCST=PXX
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV DATA    LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV DATA-Q  LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV RR      LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV RNR     LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV REJ     LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV RESET   LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV RESET-C LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV INT     LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV INT-C   LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV CLEAR   LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV CLEAR-C LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV DIAG    LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XOT RCV PVC SUP LCST=Pxx


EACH RECORD ABOVE IS FOLLOWED BY:

FLGS1=xx FLGS2=xx PR=xx XPR=xx PS=xx UPWE=xx PKTR=xxxx CU/NA=xx/xx
LUIQCT=xxxx LUBST1/2=xxxx LUBBST1/2=xxxx LUBBPFMP/SFMP=xxxx
            LUBBCFMP=xxxx NX/XF REQ=xx/xx R/X RPLF=xx/xx

zzz.bbb.ccc.ddd(ppppp) MCH name VC addr LU name LU addr XMIT COMP,
                       BFR=xxxxxxxx TYP=xx CC=xx LCST=Pxx
zzz.bbb.ccc.ddd(ppppp) CALL REQ TOMCH xxxxxxxx
zzz.bbb.ccc.ddd(ppppp) PVC SETU TOMCH xxxxxxxx
zzz.bbb.ccc.ddd(ppppp) CALL REQ TOMCH xxxxxxxx FAILED, CLEAR DIAG=xx
```

Logical Unit SYSPRINT Trace Records

LU SYSPRINT Trace Records

```
RMT name VC addr LU name LU addr RPL=xxxxxxxx DATA     SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx LUSTAT   SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx SIGNAL   SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx CANCEL   SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx RTR      SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx SHTC     SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx UNBIND   SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx REQSESS  SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx SHUTD    SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx SETLOGON SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx RESETSR  SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx OPNSEC   SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx SESSIONC SENT TO VTAM R15=xx R0=xx
RMT name VC addr LU name LU addr RPL=xxxxxxxx SENDING +RSP TO VTAM
RMT name VC addr LU name LU addr RPL=xxxxxxxx SENDING -RSP TO VTAM
```

EACH RECORD ABOVE IS FOLLOWED BY:

```
REQ/RH=xxxxxxxx HOST REQ SEQ#=xxxx SENSE=xxxxxxxx RPLURH=xxxxxx LUBST1/2 =xxxx
LUBBST1/2=xxxx
RMT name VC addr LU name LU addr  RPL=xxxxxxxx RTN name SEND RPL COMP R15=xx R0=xx
FDB2=xx
RMT name VC addr LU name LU addr  RPL=xxxxxxxx RTN name RECV RPL COMP R15=xx R0=xx
FDB2=xx
RMT name VC addr LU name LU addr  RPL=xxxxxxxx RECV SPEC (ASY) R15=xx R0=xx FDB2=xx
RMT name VC addr LU name LU addr  RPL=xxxxxxxx RECV SPEC (SYN) R15=xx R0=xx FDB2=xx
RPLAREA=xxxxxxxx RPLBUFL=xxxx RPLRLN=xxxx LURVSZ=xxxx LURVSEG=xxxxxxxx LURV
DAT=xxxxxxxx LURVEOD=xxxxxxxx
RECEIVE SPEC REQUEST: RH=xxxxxx SEQ=xxxx RPLCNTDF=xxxxxx RPLRLEN=xxxxxxxx LUBST1/
2=xxxx LUBBST1/2=xxxx
RECEIVE CHECK R15=xx R0=xx FDB2=xx
RECEIVE +RSP, SEQ=xxxx SENSE=xxxxxxxx LUBST1/2=xxxx LUBBST1/2=xxxx
RECEIVE -RSP, SEQ=xxxx SENSE=xxxxxxxx LUBST1/2=xxxx LUBBST1/2=xxxx
RECEIVE EXCP REQ, RPLFDBK2=xxxxxxxx
RMT name VC addr LU name LU addr RPL=xxxxxxxx RECV TRUNC R15=xx R0=xx
RMT name VC addr LU name LU addr RECV EXP REQ ECB xxxxxxxx REQ/RH=xxxxxx R/X
RPLF=XX/XX SEQ=xxxx
RMT name VC addr LU name LU addr RECV NOTIFY/CLEANUP CODE xxxxxxxx
TPEND FOR REMOTE addr LU name CODE=xx
REMOTE addr LU name LU addr LUIQ TIMEOUT,LUIQ BFR CT=xxxx
REMOTE addr LU name LU addr ST ERR CODE=xx LUBST1/2=xxxx LUBBST1/2=xxxx LUBBCMP=xxxx
LU name LU addr RMT name VC addr LU REL
```

Console/Trace Index

Symbols

Numerics

A

B

C

D

I

L

M

N

O

P

R

S

T

V

W

X

Y