Comm-Pro Associates
Host Network Access Support

HNAS
Guide and Reference

V2R4M0

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

This file revised November 8, 2012 10:06 am, includes maintenance thru 2400111.

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:	Direct Fax support is no longer provided
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

IBM is a registered trademark of the International Business Machines Corporation.

Cisco is a registered trademark of Cisco Systems, Inc.

Microsoft, Windows, are trademarks or registered trademarks of Microsoft Corporation.

Other company, product, and service names may be trademarks of service marks of others.

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:

<u>RFCs</u>

- RFC1613 - XOT (X25 Over TCP)

<u>Comm-Pro Associates</u>

- 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.

<u>Cisco Systems</u>

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

<u>IBM Corporation</u>

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.

HNAS V2R4Mn Table of Contents

CHAPTER 1

Introduction

This page left intentionally blank.

Chapter 1 - Introduction

This chapter describes the Comm-Pro Associates' X.25 Host Network Access Support (**HNAS**) program product. Information is provided about HNAS X.25 connectivity (NPSI components), network migration, solutions, objectives, benefits, features, and system requirements. The HNAS Documentation Overview section is provided to improve documentation navigation.

As a convention in this manual, the terms '**X.25 Host NAS**', '**Host NAS**' and '**HNAS**' are used interchangeably throughout the documentation; **router** is used to refer to the family of **Cisco XOT** (X.25 over TCP/IP) network routers (26*xx*, 36*xx*, etc.) or **IBM XTP** (X.25 through TCP/IP) network routers (2210, 2212, 2216) where XOT/XTP are the transport protocols.

Host Network Access Support

Comm-Pro's X.25 Host Network Access Support (HNAS) program product is a feature rich software package that solves a strategic problem that exists in the communications world today. Many organizations are now consolidating their SNA, X.25 and TCP/IP networks in order to optimize system operations, reduce costs, reduce system downtime and increase overall productivity. Comm-Pro's HNAS product helps to accomplish these objectives in the following ways:

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.

Provides support for native SNA terminal equipment (PU Type 1, 2 or 2.1 LEN, LU Type 0, 1, 2 or 3) using the Qualified Logical Link Control (**QLLC**) protocol over XOT. This eliminates the need for the Cisco SNA to DLSw protocol conversion feature (**new for V2R2M0**).

Note: AS/400's acting as PU Type 2.1 LENs (low entry networking nodes) have successfully run under HNAS. The remote station answers the XID request of HNAS with an XID3 response in pre-negotiation format. HNAS immediately sends SNRM which is answered by UA from the remote AS/400. The ACTPU request sent by HNAS contains ' HNAS' as an ID.

Provides support for native TTY compatible ASCII devices using the Interactive Terminal Interface (**ITI**) protocol over XTP or XOT.

Provides support for Cisco's IP to XOT translate feature which allows inbound telnet sessions to be protocol converted into ITI PCNE or PAD sessions. This allows native ASCII telnet devices (non-TN3270) to access NPSI host application's as well as the HNAS remote console subsystem.

Provides host emulation of X.25 NPSI functions thus eliminating the need for Front End Processors when they are used solely for NPSI.

Emulates the suite of NPSI interfaces for SNA and non-SNA resources.

QLLC - Support for native SNA terminal equipment allows LU Type 0, 1, 2 and 3 devices on PU Type 1, 2 or 2.1 LEN controllers to access VTAM applications using their native protocol. Terminal connections that require QLLC services use Logical Line Control Type 3 (LLC3).

In 230, QLLC callout support was added to complete the complement of callout support for all HNAS NPSI emulation components.

PCNE - Protocol Conversion for Non-SNA terminal equipment allows ITI devices to access VTAM applications as SNA 3767 LU Type 1 devices. From the host's viewpoint, there is no distinction between real LU Type 1 terminals and those that are simulated via PCNE. Terminal connections that require PCNE services use Logical Line Control Type 0 (LLC0).

IPAD - Integrated Packet Assembler/Disassembler provides host resident X.29 services for ITI devices in addition to PCNE services. Terminal connections that require IPAD (PAD=INTEG) services use Logical Line Control Type 5 (LLC5).

XPAD - Transparent Packet Assembler/Disassembler allows host applications to control PAD parameters via an action byte in the data stream. Terminal connections that require XPAD (PAD=TRANSP) services use Logical Line Control Type 5 (LLC5).

GATE - Generalized Access Transport Extension allows host applications to control X.25 virtual circuit session establishment and disconnect using a control LU session and data transfer using a separate data LU session. Terminal connections that require GATE services use Logical Line Control Type 4 (LLC4).

GATEFC - GATE Fast Connect allows host applications to control X.25 virtual circuit session establishment, disconnect and data transfer using the data LU session only. Terminal connections that require GATEFC services use Logical Line Control Type 4 (LLC4).

Provides support for GATE, GATE-FC, PCNE, IPAD and XPAD **host-to-host** application connectivity over TCP/IP based networks with no requirement for Cisco's X.25 over TCP/IP (XOT) or IBM's X.25 through TCP/IP (XTP) protocols defined in the routers. When HNAS is installed at each location end point, the hosts can communicate with each other over an IP network without any X.25 facilities or Cisco XOT routers. GATE or PCNE file transfer applications in each host can communicate with each other via the **HNAS-XOT to HNAS-XOT** connection. With this technique, packet and window sizes can be adjusted to optimize network performance, reduce CPU utilization and eliminate router serial X.25 link interfaces.

Provides transparent access to most NPSI host applications - **no modifications are required to existing host software.** HNAS ***does not*** currently support **DATE, SNI, QLLC (PUT2.1 or LU 6.2).** Please contact Comm-Pro should you have a requirement for any one of the currently unavailable NPSI products.

Provides the full scope of X.25, X.3 and X.28 functions as implemented in the XOT environment (specifically in the Cisco implementation) with the following exceptions:

- X.25 D bit (d-bit) delivery confirmation support. This is one of the general information identifier bits located in the packet information frame. It is used to set either local or end-to-end packet layer delivery acknowledgement. This support was seldom used in X.25 environments due to vastly improved networking delivery capabilities (including line quality) as well as introduction of application level (LU-LU) delivery confirmation.

- X.25 XCM (extended control mode) X25 Modulo 128 support. HNAS currently only support NCM (normal control mode) X25 modulo 8. The XCM support was initially implemented by vendors for satellite transmissions to allow more packets in flight although seldom used these day. Most X.25 networks never did or no longer support a window size greater than 7 while a window sizes of 2/4 seems to be the standard. We have found that the XOT implementation operates more efficiently with larger packets which reduces the number of packets required for transaction processing.

Network Migration

Due to the proliferation of ‘legacy’ host applications which require X.25 connectivity, many z/OS, OS/390, MVS and VM users still rely on traditional Front End Processors and the NCP and NPSI program products that run in them. The family of IBM and/or Cisco routers now offer a partial solution to bridging X.25 and TCP/IP based networks.

* **SNA X.25 DTE Support - QLLC Router Based DLSw Solution (Non-HNAS**)
  For IBM and Cisco routers, native SNA connectivity using the Qualified Logical Link Control (QLLC) X.25 transport protocol. Allows DTE to DTE and DTE to host access.

* **Non-SNA X.25 DTE Support - Partial Router Based Solution (Non-HNAS)**
  For IBM or Cisco routers, non-SNA connectivity using the X.25 over TCP/IP (XTP or XOT) encapsulation protocol. Allows DTE to DTE ***but not*** DTE to host access without the use of additional non-native equipment.

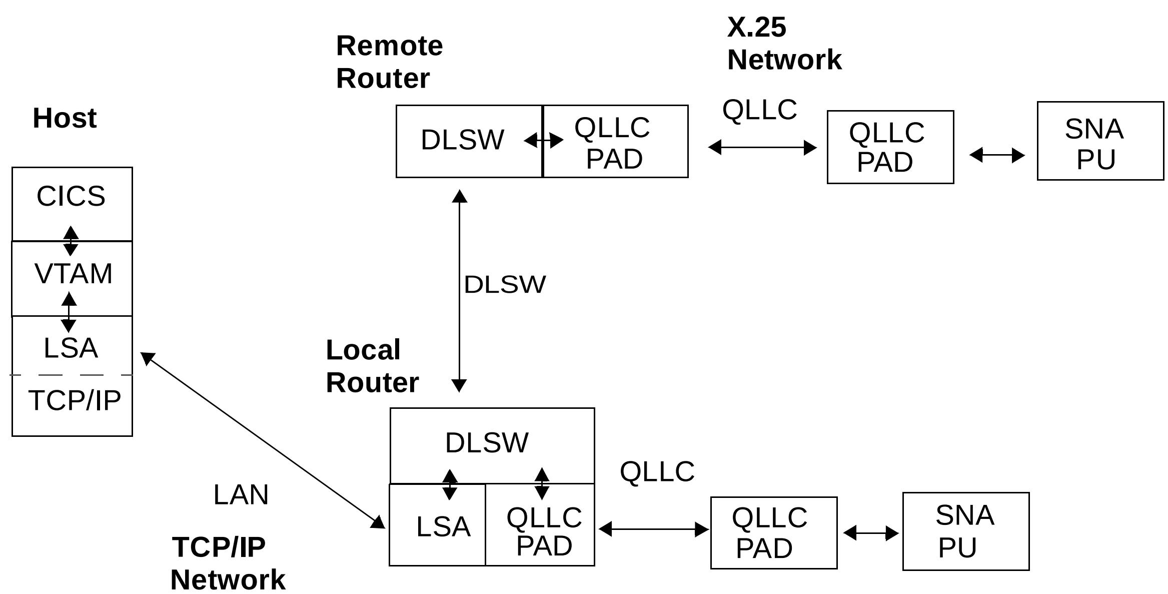
SNA X.25 DTE Support - QLLC Router Based DLSw Solution (Non-HNAS)

Native SNA X.25 DTEs connect to IBM or Cisco routers using the Qualified Logical Link Control (QLLC) transport protocol. The routers, acting as QLLC PADs, extract the SNA data from the QLLC packets then passes it across the TCP/IP network using the Data Link Switching (DLSw) transport protocol. A peer router extracts the encapsulated SNA from the DLSw packets and forwards it on to a target DTE.

For SNA X.25 DTEs, there must always be a peer DLSw router to encapsulate and de-encapsulate the SNA data.

If the target DTE is a peer SNA X.25 DTE, the peer router forwards the SNA data across the X.25 network using the QLLC protocol.

If the target DTE is a host DCE rather than a peer SNA X.25 DTE, the peer router passes native SNA traffic directly to VTAM over a Link Services Architecture (LSA) adapter (see figure 1 below).



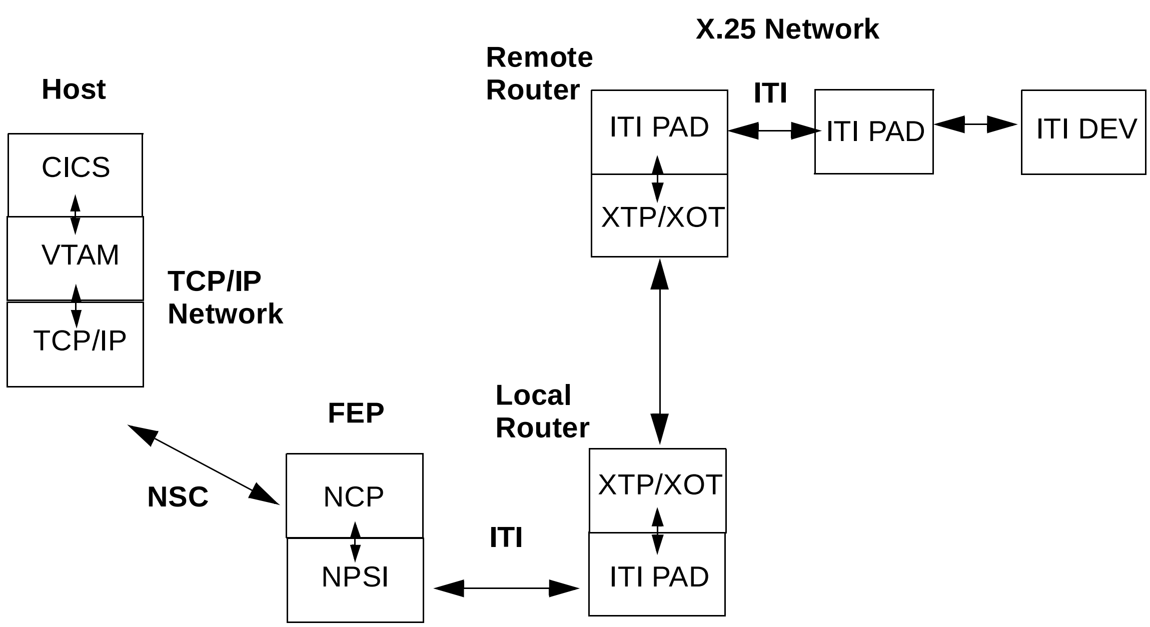
SNA device traffic is carried across the X.25 network in QLLC packets. An IBM or Cisco router extracts the SNA data from the QLLC packets then passes it to the TCP/IP network in DLSw packets. A peer router determines if the target is a host DCE or another SNA X.25 DTE. If the target is an SNA X.25 DTE, the SNA data is encapsulated in QLLC before it is passed to the X.25 network. If the target is a host, the router passes the native SNA data directly to VTAM over a LAN attached LSA adapter.

Figure 1: Native SNA traffic in a hybrid X.25-TCP/IP Network

Non-SNA X.25 DTE Support - Partial Router Based Solution (Non-HNAS)

Non-SNA X.25 DTEs connect to IBM or Cisco routers using the ITI transport protocol. The routers then propagate the ITI traffic across the TCP/IP network using the XTP or XOT transport protocol. A peer router extracts the encapsulated ITI packets from the XTP or XOT packets and forwards them on to a peer non-SNA X.25 DTE.

For non-SNA X.25 DTEs, there must always be a peer router to extract the ITI packets from the XTP or XOT packets in order to pass them on to the peer non-SNA X.25 DTE. **The routers do not by themselves provide direct connection to the host.** A FEP, with resident NPSI software, is still required (see figure 2 below).



Non-SNA device traffic is carried across the X.25 network in ITI packets. An IBM or Cisco router encapsulates the ITI packets in XTP or XOT before they are passed to the TCP/IP network. A peer router performs the opposite function and passes the extracted ITI packets on to a NPSI resident FEP. NPSI converts the ITI packets to SNA PIUs and passes them on to NCP. NCP then transfers the SNA data to VTAM over its Native Subchannel. Note that this figure is provided to show a router's ITI PAD capabilities only. Cost considerations will ultimately dictate whether a TCP/IP network or native X.25 network would be the best connection between an ITI DTE and a NPSI FEP.

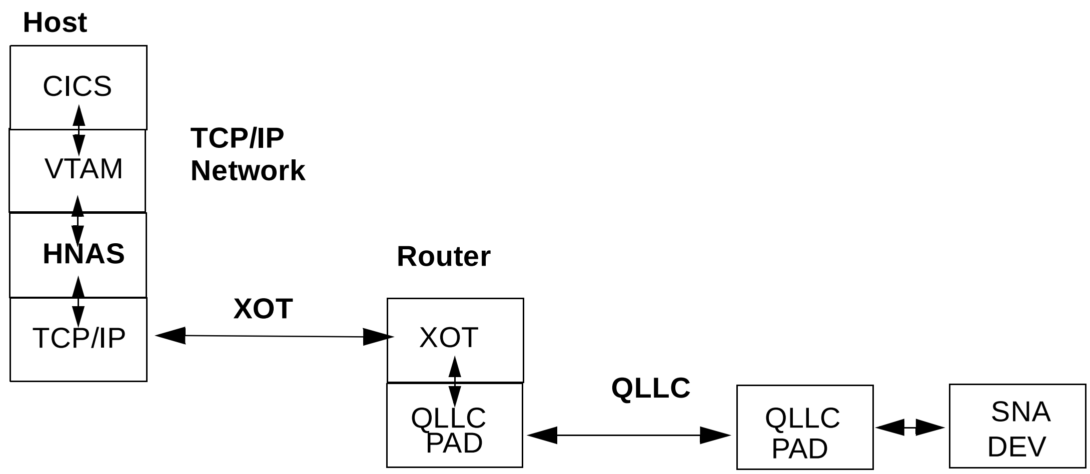
Figure 2: Non-SNA traffic in a hybrid X.25-TCP/IP Network

Solutions Using Host NAS and Routers to replace NCP/NPSI and FEPs

The missing piece in the network migration scenario is support for a router to host connection which eliminates the need for a FEP but which incorporates the NPSI functions for SNA and non-SNA equipment (QLLC, PCNE, IPAD, XPAD, GATE and GATEFC).
.
Comm-Pro's mainframe resident **X.25 Host NAS** program product provides the missing interfaces precisely where the NPSI host applications execute. Using the XTP or XOT transport protocol within a z/OS, OS/390 and MVS environment to channel attached or LAN attached IBM and/or Cisco routers, X.25 Host NAS can, in most cases, eliminate the need for Front End Processors and their associated NCP and NPSI software licenses (see figures 3 and 4 below)

Solutions using Host NAS for SNA X.25 DTE Support (Cisco routers only)

Native SNA X.25 DTEs connect to Cisco routers using the Qualified Logical Link Control (QLLC) transport protocol. The routers encapsulate the QLLC packets in XOT packets then passes them across the TCP/IP network to X.25 Host NAS. For SNA X.25 DTEs, X.25 Host NAS simulates VTAM Systems Services Control Point (SSCP) functions for PU Type 1 and 2, LU Type 0, 1, 2 and 3 equipment.



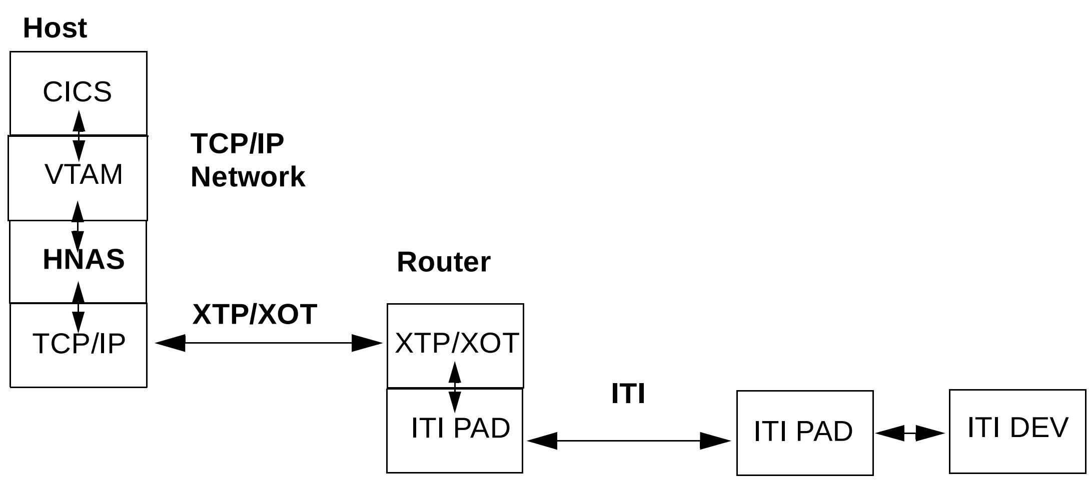
SNA device traffic is carried across the X.25 network in QLLC packets. A Cisco router encapsulates the QLLC packets in XOT packets then passes them to the TCP/IP network. The peer TCP/IP companion in this case is a host TCP/IP stack with an X.25 Host NAS component. Together, they take the role of peer router. The XOT packets are passed to X.25 Host NAS from the TCP/IP stack. X.25 Host NAS extracts the SNA data from the XOT packets and passes it to VTAM.

Figure 3: Native SNA traffic in a hybrid X.25-TCP/IP Network with Host NAS

Solutions using Host NAS for Non-SNA X.25 DTE Support

Non-SNA X.25 DTEs connect to IBM or Cisco routers using the Interactive Terminal Interface (ITI) transport protocol. The routers encapsulate the ITI packets in XTP or XOT packets then passes them across the TCP/IP network to X.25 Host NAS.

For non-SNA X.25 DTEs, X.25 Host NAS simulates LU Type 1 (SNA 3767) processing for the native ASCII terminal equipment.

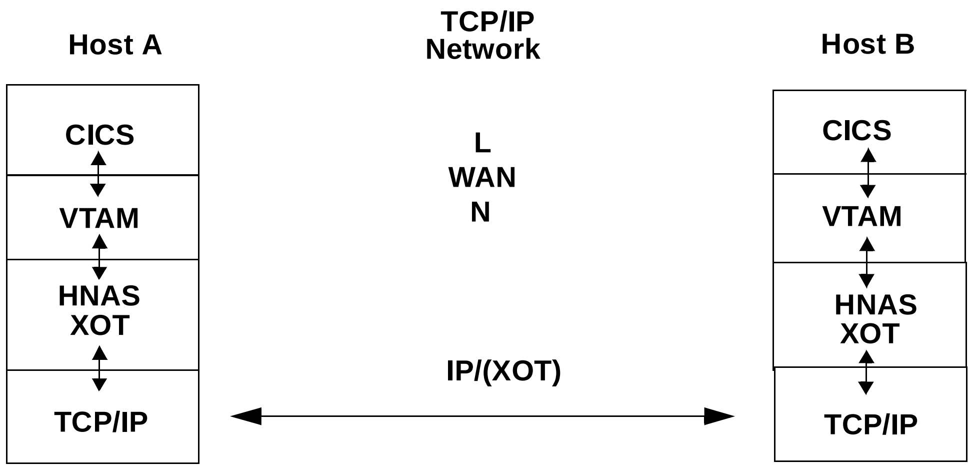


Non-SNA device traffic is carried across the X.25 network in ITI packets. An IBM or Cisco router encapsulates the ITI packets in XTP or XOT packets then passes them to the TCP/IP network. The peer TCP/IP companion in this case is a host TCP/IP stack with an X.25 Host NAS component. Together, they take the role of peer router. The XTP or XOT packets are passed to X.25 Host NAS from the TCP/IP stack. X.25 Host NAS extracts the ITI data from the XTP or XOT packets then protocol converts the ITI data to LU Type 1 (SNA 3767) so that SNA data is passed to VTAM.

Figure 4: Non-SNA traffic in a hybrid X.25-TCP/IP network with HNAS

Solutions using Host NAS for Non-SNA X.25 Host-to-Host

Host application traffic is encapsulated into XOT packets then delivered to the TCP/IP network for routing to the destination host. Cisco XOT or IBM XTP protocol services are not required in the routers because there are no physical X.25 interfaces or service requirements in the network.



Host NAS can also be used in environments with host-to-host (application to application) sessions without the requirement of XOT or XTP router protocol conversion when HNAS is used at both host end points. The HNAS XOT host messages are transported across the TCPIP network without any physical X.25 service requirements in the network or routers. HNAS eliminates the need for Front End Processors (their associated NCP and NPSI software licenses), physical X.25 networks or services and routers with XOT or XTP protocol requirements.

Figure 5: Non-SNA traffic across a TCP/IP network with host-to-host HNAS-XOT support

Objectives and Benefits

* **Elimination of Front End Processors**.
HNAS runs on the mainframe obviating the need for FEPs that are used solely to support native NPSI functions. IBM and/or Cisco routers, when used with HNAS, provides X.25 connectivity for SNA and non-SNA devices into VTAM applications.

* **Elimination of NCP and NPSI software licenses**.
No FEPs means no FEP software or hardware maintenance fee's.

* **Simplified configuration procedure**.
No Sysgen is required to generate HNAS resources. A small configuration data file is all that is necessary. This file is interpreted by HNAS when it is started. All required control blocks are created dynamically based on information supplied in the configuration data file.

* **Better use of staffing resources**.
Data center personnel no longer have to support and maintain complex NCP and NPSI Sysgens.

* **A seamless backbone network**.
Native TCP/IP traffic is delivered directly to the host mainframe.

* **Simplified management of resources**.
The Simple Network Management Protocol (SNMP), which is native to TCP/IP, can be used to manage and monitor XTP or XOT router sessions.

* **Improved network resource scalability**.
Use of the cost effective family of IBM and/or Cisco routers as FEP replacements provide scalability that can not easily be matched using Front End Processors.

* **Redundant routers reduce network downtime**.
Due to their modest cost, multiple routers can be used to provide alternate paths to the same target DTE.

* **More bandwidth for less money**.
Perhaps the most significant aspect of the HNAS to IBM router implementation is the fact that Levels 1, 2 and 3 of the X.25 protocol stack are managed by the IBM routers themselves. This means that all flow control and packet sequencing is done locally by the routers so that only data packets and session control packets need traverse the TCP/IP network. Cisco routers provide Levels 1 and 2 of the X.25 protocol stack and even though HNAS must provide Level 3 support, it still represents a significant savings in host CPU resources. This results in an increase in real bandwidth and minimizes the amount of work the HNAS must do to maintain an X.25 virtual circuit session. Other X.25 host to router implementations require the host software to perform the X.25 Level 2 and 3 protocol. These implementations consume far greater host CPU cycles. To increase bandwidth even more and improve data delivery performance, we recommend that Cisco XOT users increase the session level packet size and window size for all virtual circuits.

* **Simplified troubleshooting**.
  Diagnostic capabilities are provided by HNAS in addition to standard GTF facilities. HNAS includes extensive trace and error reporting facilities as well as a console subsystem. The console subsystem can be accessed locally via a WTOR or MODIFY interface, by line mode devices on the local router or remotely using ITI PAD or telnet access. Remotely connected console sessions are password protected for added security. For more information on the HNAS console subsystem, please refer to Console Subsystem documentation. For more information on remote console access, please refer to the description of the BUILD and REMOTE definition statements in Chapters 3 and 4.

Introduction

Features

* **Support for Cisco's XOT and IBM's XTP transport protocols**.
No changes are required to the X.25 network and complete transparency is maintained between remote DTEs and host applications. Routers can be directly attached to the mainframe channel or can be on a LAN shared by multiple routers and hosts.

* **Support of standard NPSI SNA and non-SNA interfaces**.
QLLC devices (LU Type 0, 1, 2 and 3) are supported in their native mode. PCNE, IPAD, XPAD, GATE and GATEFC sessions access host applications as LU Type 1 resources. No changes are required to host applications.

* **Elimination of NCP/NPSI Sysgens**.
A small configuration definition file and a VTAM application definition file are all that are required for HNAS operation. An HNAS **Configuration Data File (CDF)** is used to describe the router network. The CDF is interpreted by HNAS when it is started. An HNAS **Application Major Node File (AMNF)** supplies application SLU names. The AMNF provides a function similar to the NPSI Switched Major Node File and is used to identify all the SLUs that HNAS will be supporting.

  The AMNF can be produced automatically by HNAS during **FASTRUN** processing. This guarantees that the AMNF and CDF are in complete agreement and eliminates the possibility of an accidental mismatch that can occur when the AMNF is created manually.

* **Configuration error summary report**.
A report is now produced at the end of the CDF processing that lists the number of **I**nformational (RC=0 level), **D**efault and **W**arning (RC=4 level), **E**rror (RC=8 level) and **S**evere Error (RC=12 level) messages that were generated.

* **Memory requirement summary report**.
The **REGION** size requirement for a specific configuration is computed for you automatically during **FASTRUN** processing. A report is produced and logged in SYSPRINT so you will know the REGION size value that HNAS will require for normal execution. In addition to the REGION size, the report will also list the storage required for the HNAS load module (executable code), buffer pool, trace table and all control blocks.

* **Automatic application major node file (AMNF) generation**.
The AMNF can be produced automatically by HNAS during **FASTRUN** processing. This guarantees that the AMNF and CDF are in complete agreement and eliminates the possibility of an accidental mismatch that can occur when the AMNF is created manually. The AMNF is generated when HNAS is started with **PARM=FASTRUN**, a **//MAJNODE DD statement** is present and the configuration return code is 4 or less. The //MAJNODE DD statement must represent a sequential file or a member of a partitioned dataset. HNAS will produce an APPL statement for every SLU identified in the LUNAME, PVC, SVC0, SVC4 and SVC5 operands for every REMOTE definition in the CDF.

* **VTAM operand propagation to AMNF**.
HNAS has been modified to treat every non-HNAS operand that is coded on a

REMOTE definition statement as a VTAM operand. These operands are not checked for proper syntax or valid values and are propagated to the AMNF APPL statements 'asis' during **FASTRUN** processing. For example, if MODETAB=MYMODTAB is coded on a REMOTE definition statement, it will be propagated to the APPL statements associated with the REMOTE as specified since it is not an HNAS operand.

* **High Memory Support**.
HNAS can allocate its control blocks above the 16MB boundary to allow large configurations. This support is enabled when **APFXEQ** is coded in the PARM= operand on the HNAS EXEC statement. This new support allows HNAS to allocate free memory from up to 7 subpools using the **APFMEMSP=** suboperand of the PARM= operand. When APFXEQ is coded and APFMEMSP= is omitted, memory subpool 230 is assumed. If APFMEMSP= is specified, its list values are processed left to right. If memory in the first specified subpool becomes exhausted as control blocks are allocated, the next subpool is used. This continues until the end of the list is reached. If all subpools in the list are used before all control blocks are allocated, the low memory area below 16MB is used. For example, if APFMEMSP=(229,230) is specified, memory is allocated from subpool 229 before any is allocated from subpool 230 then from subpool 230 before any in the low memory area. In order to use system high memory subpools, HNAS must be link edited with the **AC=1** option and stowed in a **Authorized Program Facility (APF)** registered dataset. The standard HNAS link library can be made APF registered by placing its name in the **LNKLSTxx** member in SYS1.PARMLIB.

* **QLLC (LLC3) protocol support over XOT**.
Support for native SNA terminal equipment (PU Type 1 or 2, LU Type 0, 1, 2 or 3) using the Qualified Logical Link Control (**QLLC**) protocol over XOT has been added. This new support eliminates the need for the Cisco SNA to DLSw protocol conversion feature.

* **Simulated MCH support for GATE (LLC4) sessions**.
The HNAS configuration allows the X.25 multi-channel links (MCHs) on IBM routers to be individually addressable. This means that a host CTCP application's view of the X.25 network does not have to change. HNAS, together with the IBM router(s), become an 'extension chord' to the X.25 network.

* **SUBD= allowed on non-Fast Connect GATE MCHs**
This permits inbound GATE sessions to select a CTCP using subaddress digits in the call request packet. Previous logic allowed SUBD= only for Fast Connect GATE sessions (NPSI restriction).

* **GATE SLU activation enhancement (REMOTE OPTIONS=MCHTMR=*sec*)**.
HNAS now permits GATE SLU activation delay to be configured. In prior releases this delay was set to one (1) minute. HNAS issues an OPEN ACB and REQSESS to VTAM when the delay expires to force a BIND form the CTCP. In some cases, this delay turned out to be too long so that inbound calls would be cleared because the GATE control session SLU was not bound. The new HNAS logic allows the delay to be fine tuned to values between 4 and 60 seconds ensuring that the control session SLU will be bound by the CTCP in a timely manner. Note that the new MCHTMR value also triggers HNAS to query all data SLUs to ensure that their Application Control Blocks (ACBs) are OPEN to accept BIND requests. Each ACB the HNAS opens requires a corresponding APPL statement in the HNAS Application Major Node File (AMNF).

To force HNAS to use a shorter activation delay, specify OPTIONS=MCHTMR=*seconds*,GATE=GENERAL on the TYPE=XTP|MCH REMOTE definition statement.

* **GATE callin resource ID enhancement (REMOTE OPTIONS=LCN0USED)**.
HNAS now permits the GATE Resource Identifier (RESID) to start at zero (0) rather than one (1) plus the number of PVCs configured. This is required for CTCPs (CSFI for example) that know the NPSI LCN0=USED operand. To force HNAS to start the RESID numbering at zero, specify OPTIONS=LCN0USED,GATE=GENERAL,PVC=NONE on the TYPE=XTP|MCH REMOTE definition statement.

* **Call setup processing enhancement (REMOTE OPTIONS=ECHOFAC)**.
HNAS can now be configured to echo the facilities from an inbound Call Request packet in the outbound Call Accept packet. Some networks do not support this feature while others can use the response to further negotiate facilities values. To force HNAS to return the facilities from the inbound Call Request packet in the outbound Call Accept packet, specify OPTIONS=ECHOFAC on the TYPE=MCH REMOTE definition statement. Note that this processing is valid for non-GATE XOT VCs only.

* **Selectable SLU support for PCNE (LLC0) and PAD (LLC5) sessions**.
Each PCNE and PAD SVC connection can be mapped to a unique SLU based on the ***calling*** DTE address from the inbound Call Request packet. The HNAS configuration allows remote DTE addresses to be associated with SLUs so that only matching DTE addresses can use the SLUs. This guarantees that a host application will always ‘know’ who a remote DTE is by its unique SLU name. This feature is important for security conscious applications like CICS.

* **SLU selection via CUD IDNUM value for PCNE and PAD sessions**.
Each PCNE and PAD SVC connection can be mapped to a unique SLU based on the **IDNUM** value carried in the Call User Data field of an inbound Call Request packet. The HNAS configuration allows IDNUM values to be associated with SLUs so that only matching IDNUM values can use the SLUs. This is accomplished by specifying a hex IDNUM value (**X***idnum*) in place of decimal DTE address value (*dteaddr*) for an SVC0= or SVC5= operand entry. The following example illustrates how to configure this feature:

```
MCH1XOT REMOTE TYPE=MCH          ; logical MCH
               SVC0=(10,         ; reserve 10 PCNE SLUs
               PCNE01/X900A10I, ; map IDNUM=900A1 to PCNE01
               :
```

* **SLU/PLU fixed connection support for PCNE and PAD sessions**.
A PCNE or PAD SLU that is selected from an inbound Call Request packet can be dedicated to a single application PLU without end user input. This is accomplished by specifying an APPLNAME= parameter index after the ‘**I**’ character that delimits the *dteaddr* or **X***idnum* value for an SVC0= or SVC5= operand entry. The following example illustrates how to configure this feature:

```
MCH1XOT REMOTE TYPE=MCH          ; logical MCH
               SVC0=(10,         ; reserve 10 PCNE SLUs
```

```
PCNE01/X900A10I1, ; wire PCNE01 to CICS
:
APPLNAME=(TSO,CICS,MCHSOL)   ; supported apps
```

* **Callout support for PCNE and PAD sessions**.
PCNE and PAD SVC connections can be initiated by HNAS. HNAS simulates the NPSI CONNOUT function for switched LU resources without a Switched Major Node component. The HNAS configuration allows remote DTE addresses to be specified so that HNAS can initiate SVC sessions when a BIND is issued to the associated SLU resources. A BIND, rather than a CONNOUT, causes an XTP or XOT Call Request packet to be transferred to the remote DTE. This feature is important for host-to-host applications.

* **Alternate DTE address callout retry support for PCNE and PAD sessions**.
A PCNE or PAD SLU that is used for callout can be associated with up to 3 DTE addresses. The HNAS configuration allows up to 3 DTE addresses to be specified after the SLU name for a callout SVC0= or SVC5= operand entry (**O** delimiting the DTE addresses implies callout). If multiple DTE addresses are specified, they must be separated by hyphens as follows:

SVC5=(*vclmt*,*sluname*/{*dteaddr*1[-*dteaddr*2[-*dteaddr*3]]|**O**/*mxtname*/*cud*,...

The *cud* value, if specified, is placed in the Call User Data field of the outgoing Call Request packet. This *cud* value overrides the CUD= operand value from the same REMOTE and from the TYPE=MXT REMOTE associated with the SVC0= or SVC5= operand entry. The purpose in providing multiple DTE addresses is to allow HNAS to retry failed outbound call requests. For example, if 3 DTE addresses are specified for an SLU, the first will be used when the SLU is initially BOUND by the PLU. A Call Request packet will be created using the **first** DTE address as the ***called*** DTE address. If HNAS receives a Clear Request packet as a response to the Call Request packet, a new call will be attempted using the second DTE address. If this call also fails, a new call will be attempted using the third DTE address. If this call also fails, the PLU is notified via an **UNBIND** request. If a Call Accepted packet is returned as a response to any of the Call Request packets, normal processing continues.

Note that the RTEOUT operand on the associated HOME LOCAL definition statement must provide the proper mapping for all DTE addresses used by the REMOTE.

* **Callout connection routing via called or calling address (LOCAL RTEOUT=)**.
For Cisco servers, this feature permits outbound calls to be routed using either the ***called*** or ***calling*** DTE address from the outbound Call Request packet. The HNAS configuration now allows a direction indicator to be associated with the DTE address for each RTEOUT= operand entry as follows:

RTEOUT=(*rmtname*/*dteaddr*||{**T**|**S**},...

The **T** or **S** that delimits the *dteaddr* for a RTEOUT= operand entry indicates whether the **T**arget DTE address (***called***) or **S**ource DTE address (***calling***) will be used as the routing argument.

For PCNE (LLC0) and PAD (LLC5) calls, the ***called*** DTE address comes from the SVC0= or SVC5= operand entry, respectively, on the REMOTE definition statement. The ***calling*** DTE address comes from the DCEADDR= operand on the same REMOTE or from the TYPE=MXT REMOTE associated with the SVC0= or SVC5= operand entry.

For GATE (LLC4) calls, both the ***called*** and ***calling*** DTE addresses are in the Call Request packet that comes from the CTCP.

* **Callout connection balancing (LOCAL OPTIONS=BALANCERTEOUT)**.
For Cisco servers, this feature provides a form of round robin connection balancing across multiple routers for callout initiated connections. Specifying OPTIONS=BALANCERTEOUT on a TYPE=XOT LOCAL definition statement causes HNAS to select a different REMOTE with *no* DTE address from the RTEOUT= operand list each time a callout request is processed.

* **VTAM data transfer enhancement (BUILD OPTIONS=LUBLTCNT=*value*)**.
HNAS now permits the number of slots in the VTAM buffer list table for an SLU to be configured. Each slot in the VTAM buffer list table contains a data address and data count. This allows transfers of non-contiguous data such as that contained in an Mbit (m-bit) chained packet sequence.

In prior releases this value was set to 40 (V2R1) or 30 (V1). For most transactions, the old values were sufficient. However, for some transactions that require long buffer chains, the old values are insufficient.

The default VTAM buffer list table count is now 40. To force HNAS to use a different value, specify OPTIONS=LURBLCNT=*value* on the BUILD definition statement. You may specify a *value* between 25 and 500.

* **z/OS V1R13 TCPIP stack support**.
HNAS is now upward compatible with z/OS V1R2 thru V1R13) and is downward compatible with z/OS V1R1 and OS/390 V2R10 and below.

* **Multiple TCPIP stack support (LOCAL TCPNAME=*stkname*)**.
A single copy of HNAS can now communicate with more than one TCPIP stack. This can be useful in environments where one stack is used for testing and another is used for production and both need HNAS connectivity. The TCPNAME= operand has been added to the LOCAL definition statement to allow different server components to be connected to different stacks. If the TCPNAME= operand is not specified on the LOCAL definition statement, the value from the same operand on the BUILD definition statement is used. If TCPNAME= is also omitted on the BUILD definition statement, a value of TCPIP is assumed as the stack name.

* **Multiple ‘same type’ server support (REMOTE HOME=*lclname*)**.
A single copy of HNAS can now support multiple LOCAL definition statements of the same type. This feature allows multiple server HOME IP addresses to be specified for the same PORT number. The servers can be connected to the same TCPIP stack or different TCPIP stacks. The HOME= operand has been added to the REMOTE definition statement to allow different server components to be connected to different clients. If the HOME= operand is not specified on the REMOTE definition statement for a router,

the server accepting the socket connection will be assigned as the HOME server for the router when the socket connection is established. When multiple 'same type' server support is used, you must specify the **SHAREPORT** option on the **PORT** statement in the **TCPIP PROFILE** file. For example, if you need to configure multiple XOT servers in HNAS, the following PORT statement should be used:

`PORT 1998 TCP` *`hnasname`* `NOAUTOLOG SHAREPORT`

* **Elimination of 2000 socket limit (LOCAL SOCLMT=*value*)**.
Each HNAS server component may now support up to 65503 client sockets (HNAS theoretical limit). The SOCLMT= operand has been added to the LOCAL definition statement to allow the server socket limit to be specified. If the SOCLMT= operand is not specified on a LOCAL definition statement, a value of 2000 is assumed. If a SOCLMT= value greater than 2000 is specified, multiple server component Task Information Elements (TIEs) are created for the same server HOME IP address. The number of TIEs that are created is computed from SOCLMT/2000+1. HNAS will LISTEN for client connections on each TIE that is created. In this case, the TCPIP stack will present connections to HNAS on each TIE in a round robin fashion in order to load balance the connections across all the TIEs for a specific HOME IP address.

* **Dynamic IP address assignment (REMOTE IPADDR=DYNAMIC)**.
For Cisco routers, this feature eliminates the need to pre-define router IP addresses in the HNAS configuration for inbound connections. Specifying IPADDR=DYNAMIC and PORT=DYNAMIC on a TYPE=XOT REMOTE definition statement creates a pool of TCPIP sockets that are reserved for inbound XOT connections only. The number of sockets in the pool is determined by the VCLMT= operand. The IP address for a session is set when the socket connection is established rather than at HNAS configuration time. This eliminates the need to configure every router in the network for inbound connections and allows routers to be added to the network without having to modify the HNAS Configuration Data File.

* **Shared socket support (REMOTE IPADDR=*a.b.c.d*,PORT=1998)**.
For Cisco routers, this feature eliminates the need to pre-define separate pools of inbound and outbound TCPIP sockets for a specific router in the HNAS configuration. Specifying IPADDR=*a.b.c.d*,PORT=1998 on a TYPE=XOT REMOTE definition statement (which identifies a specific router) creates a single pool of TCPIP sockets that can be used for both inbound and outbound sessions. The number of sockets in the pool is determined by the VCLMT= operand. A socket from the pool is allocated to a activating session on a first come, first served basis. The same socket can be used for an inbound initiated connection at one time then an outbound initiated connection at another time.

* **TCPIP buffer utilization enhancement (REMOTE OPTIONS=TCPRBLMT=*value*)**.
HNAS now permits the number of buffers that are staged to receive input on a socket to be configured. In prior releases, this value was set to seven (7). For XTP sockets, this was a reasonable number because all VCs are multiplexed across a single TCPIP socket. For XOT VCs, this value was excessive (in most cases) because each VC uses a separate TCPIP socket. We have observed that for XOT VCs, a value of two (2) is adequate for a variety of transactions. Configuration environments with hundreds of XOT VC connections will benefit from the reduced memory requirement that results from using a smaller staging buffer count.

The default staging buffer counts are 7 for XTP and 2 for XOT. To force HNAS to use a different value, specify OPTIONS=TCPRBLMT=*value* on the TYPE=XTP|XOT REMOTE definition statement. You may specify a *value* between 1 and 7.

* **TAP (keep alive) processing enhancement (REMOTE TAP=*value*)**.
For Cisco routers, HNAS now uses a special XOT Call Request packet as the keep alive request. In prior releases, an XOT Clear Request packet was used. This led to problems because the IOS level in some routers did not return a Clear Confirm packet to the HNAS Clear Request packet which is the required keep alive response. This caused the HNAS keep alive response timer to expire and after a second failed attempt, the router to be taken off line. For this reason, the default TAP value for a TYPE=XOT REMOTE definition statement was changed from 10 (seconds) to zero (0) to inhibit keep alive processing. Since HNAS now uses a Call Request packet as the XOT keep alive request, we expect all routers to respond regardless of the IOS level.

The default TAP value for a TYPE=XTP REMOTE definition statement remains at 120 (seconds). The XTP keep alive request is, as it has always been, the standard KEEP ALIVE packet as defined by the XTP specification.

* **Audit logging of connections**.
XTP and XOT connection initiation and termination requests are noted in a user defined log file.

* **Console subsystem**.
HNAS provides a console interface that can be accessed locally, via a standard host WTOR|MODIFY interface, or remotely via an ITI PAD or Cisco router XOT ITI connection. Remote access is password protected. A user selectable X.25 ITI session can originate using a remote PAD, telnet to Cisco PAD translate function or via local Cisco AUX or CON router PAD session support.

The console subsystem provides a mechanism to start, stop and display trace information, start, stop and display statistics information, display and alter HNAS storage areas, display and alter LOCAL and REMOTE configuration information, display router state information, display 'simulated' NPSI Multi-Channel Link (MCH), Virtual Circuit (VC) and Logical Unit (LU) state information and monitor HNAS resource utilization.

* The **PING XOT console command** has been added to the console subsystem that will allow the console operator to check on the existence of a router on the IP network as well as a specific MCH on the router. The HNAS PING command is not a traditional IP PING command but rather an X.25 PING. The HNAS PING command requires an IP address to identify a router and an X.25 DTE address to identify the MCH link. These two pieces of information allow HNAS to establish an IP connection to the router and then an X.25 (XOT) connection via the MCH link. An indication of success or failure for the PING request is reported back to the console operator.

* The **MRMT console command** has been enhanced to allow users to modify SLU definitions in the LUNAME= operand of TYPE=SPU REMOTE definition statements via privileged local or remote console access as follows:

* **WTO CART= support for session manager (Netview, TDSLink) operator consoles.** The CART= operand added to all WTO macro to pass a token from MODIFY command back to the MODIFY sender in order to associate console commands with responses. This is used by TDSLink software to solicit specific information from the HNAS console subsystem.

  The new ALRMCART= operand has been added to the BUILD definition statement so that WTO alarm messages can be routed to a specific component like TDSLink. The ALRMCART= token is passed to the WTO service routine via the CART= operand for alarm messages only.

* **Alarm filtering support (BUILD ALRMFLTR=)**. HNAS alarm messages may now be filtered by message identifier (e.g., NAS3701W). You may specify a up to 15 alarm message IDs (wildcards (*) are accepted) plus a disposition that tells HNAS what to do when a match occurs in the filter list (accept, suppress or purge). On the BUILD definition statement, code the following:

  ```
  ALRMFLTR=(A|S|P,id1(A|S|P),...,idn(A|S|P)).
  ```

  A for accept causes the alarm message to be written to the system console and SYSPRINT. **S** for suppress causes the alarm message to be written to SYSPRINT only. **P** for purge causes the alarm message to be discarded and not written to the system console or SYSPRINT. The first ALRMFLTR suboperand is used to set the default action for an alarm message ID (idn) when the disposition is omitted.

  Alarm filter values may be added, deleted, modified or displayed using the **ALARM FILTER=** console command.

* **Alarm limiting and logging support (BUILD ALRMLMTS=)**. HNAS alarm messages that pass the filter processing may now be limited from display at the system console to reduce system WTO usage or prevent depletion of the system WTO buffer pool. You may specify a time interval during which the limits apply plus a count for the maximum number of messages of a given severity that will be displayed during the specified time interval.

  Alarms will always be logged in the system alarm logging table event if they are deleted via filtering or limiting. The alarm logging table is sorted by alarm ID and contains a count of the number of times the associated alarm was issued and a timestamp that indicates the last time the associated alarm was issued.

  The alarm logging table may be cleared (all entries removed), reset (all counters reset for current entries) or displayed using the **ALARM LOG=** console command.

  Please refer to BUILD operand ALRMLMTS= in Chapter 4 for additional information and coding instruction.

* **Remote alarm console support**. Remote console connections (RMTCONP start parameter) can be conditioned to receive the same HNAS alarm messages that the system console receives. Remote alarm console status is set using the **ALARM ME** console command. Remote consoles,

like the system console, are subject to alarm filtering and limiting which can be controlled via the **ALARM FILTER=** and **ALARM LIMITS=** console commands.

* **Multiple SYSOUT support**.
Multiple SYSOUT datasets maybe specified in the HNAS start JCL which can then be closed and opened using the **PRINT CLSOPN *ddname*** console command. This command is valid for the system console or privileged remote consoles. This feature provides a mechanism for keeping multiple trace logs separated during HNAS execution. If DISP=SHR is specified (recommended), it also allows a trace log to be captured and transferred (FTP or e-mail attachment) to Comm-Pro for problem diagnosis without having to stop HNAS.

* **DYNAMIC SYSOUT support.**

PRTSWLST=({<u>LOOP</u>|STOP},SWITCHAFTERINIT,SWITCHAT*time*,
{*ddname*1|DYNAMIC=*outclass*},...,{*ddname*n|DYNAMIC=*outclass*})

was added to the BUILD definition statement to provide 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. The DDNAMEs you specify or request dynamically are used sequentially. The default SYSPRINT file is always used initially (ddname=SYSPRINT).

* **Elimination of system console configuration messages**.
HNAS now writes all configuration messages during the CDF scan to SYSPRINT only unless you specifically request that they also be written to the system console. A new default start parameter **SHOWCNFG OFF** was added to control the configuration message function. For those installations that wish to continue seeing these messages on the system console, you can specify **SHOWCNFG** as an HNAS start parameter.

Note that the configuration messages are always written to the SYSPRINT log file unless the **ALRMFLTR=PURGE** option is specified. This BUILD definition statement operand operates independently from the **SHOWOFF** or **SHOWERR** start parameters.

The **SHOWERR** start parameter prevents HNAS from delivering **I**nformational, **D**efault and **C**omment messages to the system console. Only error messages are delivered to the system console when **SHOWERR** (the new default) is active.

* **Initialization complete notification alert**.
HNAS now issues an initialization complete message after normal completion of its activation phase. This new alert message is independent from other HNAS and router connectivity alert messages which were previously used to determine HNAS availability. The new message now provides a more accurate representation of HNAS system availability. The new message is as follows:

NAS0001I HOST NAS INITIALIZATION COMPLETE, ALL FUNCTIONS READY

Note that this message is always written to the system console as well as the SYSPRINT log file.

Introduction

* **TCPIP alert message format enhancement**.
HNAS now produces more standardized message formats for TCPIP event alert and error messages. The new format includes a descriptor of the type of TCPIP resource (server or client) as well as the configured name for the resource. Examples of the new message format are as follows:

NAS2210I SERVER=010.117.056.221(01998) SOCKID=001F PCEID=0005 NAME=*lclnm*
NAS2210I SOCKET CONNECTION CLOSED

NAS2268I CLIENT=010.117.056.100(13153) SOCKID=001E PCEID=0007 NAME=*rmtnm*
NAS2268I ACCEPTED CONNECTION PASSED

* **Extended Event Reason Code support.**
Extended event reason codes are now beginning to be displayed in various alert/alarm messages, trace entries and some console display messages. These extended codes will make it easier to define session connect and disconnect causes as well as improve general debugging.

* **Internal tracing facilities**.
HNAS provides an extensive tracing capability that allows data at the TCP/IP, VTAM, XTP, XOT, VC and LU levels to be selectively logged in a memory resident trace table. Trace entries can also be written to the HNAS log file. Tracing can be started automatically when HNAS activates using EXEC parameters or can be started dynamically via console commands.

Additional trace logic has been added to trace configuration processing (TRCCNFG), WAIT and POST events (TRCTASK) and alarm logging table updates (TRCBST). These traces can be run independently or in conjunction with existing HNAS traces. As with all HNAS tracing, the new trace logic can be started automatically when HNAS activates using EXEC parameters or can be started dynamically via console commands.

For additional information on product features please refer to the HNAS Summary of Changes & New Features (by VnRnMn level) section in Appendix D of the HNAS Guide and Reference documentation.

Customization Support

* **Customization support based on individual user requirements**.
  Comm-Pro can modify the functions provided by HNAS to suit an installation's specific requirements. Although the 'off the shelf' version of HNAS will be sufficient for most users, some may choose to add non-standard or user-defined functions. Comm-Pro has always had a policy of providing custom modifications on a time and expenses basis.

Applications and Products Supported

Following is a list of applications and products that have been successfully tested with HNAS:

*** ARBITER** (PAD) from TANGRAM.
* **Artemis** and **Multitran** with the Integrated PAD (IPAD) function.
* **AS/400** PU Type2.1 LENs (low entry networking nodes) function.
* **btx Online Banking System** (GATE, the CTCP was written by IBM while the applications were added individually by the various banking groups).
* **CECA** (**SICA**, **SIAMI** and **TAF**) applications from Spanish Savings Bank Federation.
* **CICS**, **IMS** and **TSO** (PAD and PCNE) from IBM.
* **CICS** (GATE FC) from IBM.
* **CICS** (SNA QLLC) with PU Type 2.0 for IBM 3174, IBM 4700 or IBM PERSONAL COMMUNICATIONA/3270 (PCOM).
* **Connect Express** (GATE and PCNE) (alias **TOM**) from STERLING COMMERCE.
* **Connex** supporting **POS** (GATEFC) and **ATM** transactions from eFunds.
* **CSFI** and **CSFI/VCP** IBM with the GATE function from IBM.
* **DataFono** ISARX25 support.
* **EDITRAN CICS** (GATE) function from INDRA.
* **EDITRAN IMS** (GATE) function from INDRA - See following ISARX25 entry.
* **IMS**, **CICS** and **TSO** (PAD and PCNE) from IBM.
* **IMS** (PCNE) and **CICS** (GATE FC) from IBM.
* **ISARX25 3745** NPSI replacement product in Spain can be replaced using HNAS in EDITRAN (GATE) environments. See Application and Product Notes section for additional information.
* **Multitran** and **Artemis** with the Integrated PAD (IPAD) function.
* **ONGUM** (PCNE) file transfer function.
* **OSI/FTAM** (GATE) from IBM.
* **Pelican** (now Interpel)
* **RVS File Transfer** System (GATE) from VW - GEDAS.
* **SABRE** into IMS (PCNE) from SABRE INC.
* **SIMVTAM** (SIM3270/SIMPC) from Simware.
* **STX** (Unicenter CA-STX) CICS Callout (PAD) Support.
* **SWIFT** FIN into IMS (PCNE) from SWIFT and KEYSTONE SYSTEMS.
* **TDS-ES** and **TDSLink** agents for S/390 monitoring from Telecoms Data Systems.
* **VIRTEL** (GATE) from SYSPERTEC.
* **XFB.Monitor** (**CFT** et **Inter.Pel)** from Axway a Sopra Group Company (GATE and PCNE)

Application and Vendor Product Notes

This section currently contains information on recent application and vendor product specific notes. Entries follow:

ISARX25 3745 NPSI replacement product in Spain can be replaced using HNAS in **EDITRAN** IMS (GATE) and CICS (GATE) environments. Depending on the options used, HNAS may be able to support ISARX25 LLC0, LLC3, LLC4 and LLC5 devices.

If you want to migrate from ISARX25 to HNAS and your application is EDITRAN under IMS, and you are using ZCODTRA macro in ISARX25, then an IMS exit routine must be installed in IMS to replace the ZCODTRA function. This exit routine can be provided by INDRA. This routine has to be assembled and link-edited in your environment. Then, IMS MUST be generated to take into account this exit routine. The purpose of this exit routine is to add a transaction code in front of the incoming data from the X.25 network: this function is done by ISARX25. This exit routine is the same as the one which is used when NPSI is running in the 3745 instead of ISARX25.

In a **NPSI/IMS** environment a callout operation is started when a Switched Major Node is activated. In an **HNAS/IMS** environment the virtual LUs in HNAS are defined by APPL statements in an Application Major Node. There is no callout information in the this major node so activation does not cause creation of an X25 Call Request packet. To generate a call request take the following steps:

1. Define the HNAS LUs in IMS.
2. IMS command: START LTERM ltermname (In IMS, ltermname is the HNAS-slu-name).
3. IMS command: OPNDST NODE HNAS-slu-name.
   HNAS-slu-name is the VTAM HNAS lu defined with the VBUILD TYPE=APPL statement. The OPNDST causes a BIND to be sent to HNAS. The BIND triggers the Call Request packet build using information in the HNAS CDF.

The above commands can be done via IMS auto operator or via System Automation.

ISARX25 differs from **NPSI** in the way that M-bit chains are presented to the PLU when GATE is used:

ISARX25 - an M-bit chain is delivered to the PLU as an RU chain (FIC/MIC/LIC) using the SLU send RU size from the bind.

NPSI - an M-bit chain is delivered to the PLU as a series of OIC RUs using the SLU send size from the bind.

EDITRAN is not able to process the NPSI or HNAS OIC RU sequence. The correction is to set the SLU send RU size to a number large enough to contain the M-bit chain. This ensures that the PLU receives a single OIC RU for the M-bit chain. Be sure that MBITCHN=YES is coded on the HNAS TYPE=MCH REMOTE. We have seen cases where the HNAS REQSESS operation used to start an EDITRAN GATE data session fails with SENSE=0805000B because **OPTIONS=REQSESSDELAY=0** was not coded. The default REQSESSDELAY value (2 seconds) allows the PLU to acquire the resource at the same time HNAS issues a REQSESS that asks the PLU for a BIND.

Datafono ISARX25 support is a special order item under HNAS 240.

STX (Unicenter CA-STX) CICS Callout (PAD) Support requires custom HNAS Space and Even parity translate tables.

System Requirements

X.25 Host NAS runs in z/OS, OS/390 (MVS open edition) and MVS/ESA environments. z/OS V1R1 thru V1R13 (z/OS V1R2 thru V1R3 requires HNAS 211 and above, while z/OS V1R4 and above requires HNAS 230 and above,), OS/390 V1R1 or later and MVS/ESA V4R3 or later are currently supported. OS/390 V2R5 and earlier releases run under the IUCV interface while OS/390 V2R8 and later run under the MAPI interface.

X.25 Host NAS requires the facilities provided by the host TCP/IP stack in addition to VTAM to establish and maintain sessions between X.25 DTEs and host applications.

X.25 Host NAS establishes its presence to the host TCP/IP stack and VTAM when it is started. A TCP/IP Configuration Data File (CDF) is required to describe the router network. A VTAM Application Major Node File (AMNF) is required to define the Logical Units that will be used for host application access. IBM's TCPIP stack V2R2M1 or later and VTAM V3R4 or later are required. HNAS can also run with the Interlink TCPIP stack V2R1 or later.

X.25 HNAS is distributed via our FTP Server Access (e-distribution), on Compact Disc, or DAT tape (4mm) depending on customer requirements.

For z/OS, OS/390 and MVS, the product distribution contains a MACRO library and link edited OBJECT library in TSO XMIT format suitable for restore using the TSO RECV command. Installation JCL is provided to restore the libraries and produce a Host NAS load module.

For z/OS, OS/390 and MVS, the following additional requirements must be met.

X.25 Host NAS must execute as an **Authorized Program Facility (APF)** registered program so that its control storage can be allocated above the 16 megabyte boundary.

X.25 Host NAS must execute with the same task dispatching priority as VTAM and TCPIP. Specify **CLASS=c,PRTY=14** to set job class **c** (VTAM's job class) and the maximum dispatching priority within the job class.

X.25 Host NAS must execute in **non-swapable** mode in order to ensure that data is moved between VTAM and TCPIP in an expeditious manner. This can be accomplished when Host NAS is started as a start of task (as a procedure in SYS1.PROCLIB) or by specifying **ADDRSPC=REAL,REGION=0M** to make it **non-pageable** and to set the real storage size to dynamic. Note that the exact memory size (region size) is a function of the HNAS load module size and configuration requirement.

The **REGION** requirement for a specific configuration can be computed for you by executing Host NAS with the **FASTRUN** option prior to real execution (**new for V2R2M0**). We also **recommend that you perform a FASTRUN pass to identify any errors** in the **C**onfiguration **D**ata **F**ile prior to performing HNAS activation with the TCP/IP stack.

X.25 Host NAS must not be allowed to terminate due to time expiration. Specify **TIME=NOLIMIT** or **TIME=1440** to prevent HNAS from ABENDing with an SEC6 because it's execution time limit has elapsed.

Router Requirements

HNAS supports **Cisco** XOT enabled routers and **IBM** XTP enabled router. Interface types, memory requirements and IOS (software) levels are dependent upon configuration size as well as resource types.

Bintec XOT enabled routers, primarily used as frontend device for remote isdn and/or x.25 links (HNAS users in Germany).

TELDAT XOT enabled routers (HNAS users in Spain).

Installation Configuration Profile

Company Name	
Address 1	
Address 2	
City, State	
Zip Code, Region	
Country	
Contact:	
Telephone #	
E-mail:	
Notes	

FEP # Type	FEP System Environment	NPSI X.25 VC # Type	Routers # Type	Host System Data	Host Appl Data
__ 3745 __ 3725 __ 3720 __ 3705 __ ____ __ ____ __ ____ __ ____	NCP V_R_ PEP V_R_ NPSI V_R_	__ PCNE __ IPAD __ XPAD __ GATE __ GATEFC __ ________ __ ________	__ 2210 __ 2212 __ 2216 __ 26nn __ 36nn __ 45nn __ 7nnn __ _______	z/OS V_R_ OS/390 V_R_ MVS V_R_ TCPIP V_R_ VTAM V_R_ _____ V_R_	CICS V_R_ IMS V_R_ TSO V_R_ _____ V_R_ _____ V_R_ _____ V_R_

Figure 4: Installation Configuration Profile

CHAPTER 2

Installation and Activation Procedures

This page left intentionally blank.

Chapter 2 - Installation, Generation and Activation Procedures

This chapter describes the libraries and procedures that are used to install, generate and execute a Comm-Pro X.25 Host Network Access Support (**HNAS)** load module.

HNAS is an application program that communicates with the IBM TCP/IP stack and uses application to application VTAM sessions to move data between itself and programs that formerly communicated using NPSI.

This Chapter also provides:

- An overview of the configuration files required to operate HNAS (Configuration Data File (CDF) and VTAM Application Major Node File (AMNF)).

- A discussion of HOST environment considerations (APF, Security, RACF, TCP/IP, Netview, etc.).

- A description of the start parameters that may be passed to HNAS via the PARM= parameter on the EXEC statement used to invoke HNAS.

Following is a list of Product Component Identifiers & Installation Variables that should be reviewed to understand the terms, variables and fields associated with the installation of this product.

Product Component Identifiers & Installation Variables

Throughout this documentation section the following HNAS abbreviated product distribution terms are used:

Table 1: HNAS Component Identifiers & Installation Variables

aparid	4 digit number identifying the HNAS maintenance level - (e.g. 0147, meaning maintenance through APAR 0147 applied).
cid	3 character customer identifier assigned by Comm-Pro (e.g., cpt)
cust#	5 digit customer number assigned by Comm-Pro (e.g., 77345)
date yyyy-mm-dd	identifies the product distribution creation date (e.g., 2006-02-21)
hlq - high level qualifier	represents the user dataset name (DSN) qualifiers assigned to HNAS data sets. The lowest DSN qualifiers (e.g. HNASMAC) are specified by Comm-Pro. While any valid qualifier may be used, it is recommended that the qualifier reflect the HNAS version and maintenance level. E.g *hlq*=SYSX.@2300141 to reflect V2R3M0 with apars up to 0141 applied. Formerly QQQQ/qqqq.

Table 1: HNAS Component Identifiers & Installation Variables

host	identifies the host system type (e.g., Z/OS, OS/390 or MVS)
LNS	is the IBM registered SMP/E prefix for HNAS. HNAS component names start with this prefix when installed using SMP/E.
SHIPID	represents the distribution options and customer controls
vrm* | V*n*R*n*M*n	identifies the HNAS product release level (e.g., 230 for V2R3M0)
vrmnnnn (***aparid***)	identifies the ***vrm*** (version, release and modification) and ***nnnn*** maintenance level of the product (e.g., 2300100 for V2R3M0)
.bin .ext .str .txt .zip	See Product Distribution Media File Formats section

Installation Process

This section provides information on the HNAS product installation process, product distribution media location and types, allocation and transfer methods as well as complete installation instructions for the HNAS product.

While the majority of HNAS documentation is downward compatible with earlier releases this section is not compatible with HNAS 230 or earlier. Please refer to your specific product level guide for additional information.

Installation Types

Following are the two HNAS distribution installation types currently available for the HNAS program product:

Standard (Non-SMPE) Distribution

Non-SMP/E is our standard (default) product installation method. This is the format that the majority of companies prefer for ease of product installation as well as the application of product refreshes or upgrades for new features or maintenance support.

SMP/E Distribution

If SMP/E installation is required the HNAS distribution libraries will be in a format suitable for processing by SMP/E.   The HNAS component prefix registered with IBM is LNS. The SYSMOD ID for HNAS is LNS0vrm (vrm = HNAS version, mod, release level).

Once HNAS has been installed using SMP/E, maintenance is provided by installing refresh PTFs which replace all HNAS components so HNAS gets to a specific APAR level. Please refer to Chapter 6 (Maintenance) for information on installing PTFs using SMP/E.

Please refer to the Installation Using Non-SMP/E or Using SMP/E section for product installation instructions once you have reviewed the remainder of this section.

Product Distribution Media Location

The HNAS product is primarily provided via edistribution from our FTP server. E-mail attachment can also be provided. For customers unable to support edistribution methods or those preferring physical media delivery CD-ROM media is also available. The CD-ROM product distribution may be located is a single hnas|lns_product-level-info.zip file or a directory with a name representing the file type.

The delivery method options are provided by your HNAS sales or support organization.

Product Distribution Media File Information

The HNAS product is normally provided via a ZIP file located on the Comm-Pro FTP Server, as an E-mail attachment or on a CD-ROM. The ZIP file contains the distribution members that are required to install the product. The ZIP file name has the following format:

hnas|lns_vrmaparid_date_cust#_cid.zip -->Zip archive file containing the respective product type installation files.

hnas|lns - The distribution type - hnas for non-SMP/E and lns for SMPE
vrm - The Version, Mod and Release level of the release
aparid - The APAR maintenance level that this release was generated under
date - The date (yyyy-mm-dd) that this product distribution was generated
cust# - The 5 digit Comm-Pro customer control number
cid - The 3 character Comm-Pro customer identification

Example: hnas_2400045_2004-06-17_99000_cpt.zip

Table 1: HNAS ZIP Distribution Archive File Content

Non-SMP/E	SMP/E
n/a	lnsjcli.str
hnasmac.str	lnsmac.str
hnasmacx.str	n/a
hnasobj.str	lnsobj.str
hnasobjx.str	n/a
n/a	lnssrc.str
hnasgjob.bin	smpgjob.bin

Files are listed as related to each distribution type. n/a identifies file as not applicable to the respective zip file or distribution type.

The **hnas|lns_*product-level-info*.zip** ZIP archive distribution files contain all of the *.STR , *.BIN (Binary EBCDIC) and *.TXT (ASCII text) distribution files required to install the HNAS product.

Product Distribution Media File Formats (*.ext)

Table 2: HNAS *.*ext* File Formats

File ***.ext***	Description
*.bin	represents Binary EBCDIC files RECFM=F|FB and LRECL=80 flat files. These file types are transferred transparently (binary mode) to the host into sequential files.
*.str	represents stream files (Binary) created using the TSO XMIT (TRANSMIT) command. These files are unloaded partitioned datasets. The TSO stream files are converted back into PDS format using the TSO RECV (RECEIVE) command which is invoked as part of the HNAS product installation process.
*.txt	represents text ASCII format files that can be viewed or printed on any PC and don't require any translation. These files aren't meant for host transfer (see respective *.BIN file)
*.zip	represents zip achieve file containing the distribution files. The zip file can be transferred directly to your host if you have a host unzip product such as PKWARE's UnZip product (The zip file type is transferred transparently (binary mode).

Note: All *.STR, *.BIN and *.ZIP files *must* be transferred in binary mode.

Please refer to the specific SMP/E and Non-SMP/E installation sections in this chapter for additional file information.

Installation Checklist

The following checklist outlines what you will need to do to install the HNAS program product on your host mainframe. We recommend that you review all Checklist items and references prior to installing the product. Upon reviewing this list, it is assumed that you have already determined your product type (Non-SMP/E or SMP/E) and product delivery method (FTP, E-mail or CD-ROM media) and were advised by your HNAS sales or support representative as to the availability of the product.

- Acquire the HNAS product edistribution (FTP or E-Mail attachment) or CD-ROM media.

- Transfer or copy the hnas|lns_*product-level-info*.zip distribution product media file to a staging PC, unzip the archive file so that the distribution files are ready for transfer to your host.

  An optional method is to transfer the **zip** file directly to your host if you have a host unzip product such as PKWARE's UnZip product.

- Transfer the files from your staging PC to the host using binary FTP, IND$FILE or other host file transfer method (product install section contains DSN type and space allocation information).

- Customize and execute the HNASGJOB (non-SMP/E) or SMPGJOB (SMP/E) product installation EXECs to allocate the ***hlq***.HNASCNTL or ***hlq***.LNSCNTL PDS and load it with the jobs required to install and run HNAS. The name of the install EXEC in the .zip file is hnasgjob.bin (non-SMP/E) or smpgjob.bin (SMP/E).

- Once installation & generation steps are completed, the HNAS load module is ready for execution. Proceed to Configuration Files (CDF and AMNF creation, environment review) and the Activation and Run Time Considerations sections prior to executing product.

We recommend that you read the Installation section in it's entirety before continuing with the actual product Installation the using SMP/E or Non-SMP/E sections. Understanding the installation content will allow you to become familiar with the distribution, staging PC and Host files & libraries required to generate the product.

It's also **very important** that you review the Host Environment Considerations section for TCP/IP and Security requirements (userid DSN registration issues, APF, RACF, Top Secret, etc.) as well as the HNAS Authorization Considerations section regarding authorization distribution information for trial and registered users.

Host Distribution Libraries and Their Usage

The following table described the HNAS product source installation files and destination host files and their associations. The HNAS program product libraries are initially required for the installation process and subsequently for run time execution. Following is a description of each HNAS Host library:

Table 3: Host Distribution Libraries and Their Usage

Dataset Names	Description
Non-SMP/E	**Non-SMP/E**
HNASGJOB	REXX EXEC is used to create the ***hlq***.HNASCNTL PDS which contains the JOBs used to install and run HNAS. Prior to execution the EXEC is customized in order to provide the HNAS configuration information (***hlq***, unit and volume information, etc.).
HNASCNTL	PDS (created by HNASGJOB) containing jobs used to install, build and run HNAS. The main members are: HNASALOC Allocate HNAS data sets. HNASRCV Create libraries from .str files. HNASMNT Build or re-build HNAS load module. HNASXEQ Run HNAS
HNASLOAD	PDS containing the HNAS load module created by the installation process.
HNASMAC	PDS containing macros required for the assembly of modules with TCP/IP and VTAM dependencies during the installation process.
HNASMACX	PDS containing custom macros and/or macros that have been revised by the installation of maintenance. This library is concatenated in front of HNASMAC in the assembly ***//***SYSLIB DD statement list (see ***hlq***.HNASCNTL(NASASM)). This library may be used for the user's Configuration Data File (CDF) and the Application Major Node File (AMNF).
HNASOBJ	PDS containing preassembled HNAS modules (no TCP/IP and VTAM dependencies).
HNASOBJX	PDS containing preassembled custom user modules and/or modules revised by the installation of maintenance. Like the HNASOBJ dataset, the HNASOBJX dataset contains preassembled modules with no TCP/IP and VTAM dependencies.

Table 3: Host Distribution Libraries and Their Usage

Dataset Names	Description
SMP/E	**SMP/E**
SMPGJOB	REXX EXEC used to create the ***hlq***.SLNSCNTL PDS which contains the JOBs used to install HNAS using SMP/E. Prior to execution the EXEC is customized in order to provide the HNAS configuration information (hlq, unit and volume information, etc.).
SLNSCNTL	Created by the SMPGJOB rexx exec. The main members are: ALOCUCAT Allocate user catalog. ALOC Allocate HNAS and SMP/E data sets. SMPPROC PROC used by all SMP/E jobs. SMPRECV SMP/E RECEIVE. SMPAPLY SMP/E APPLY. SMPACPT SMP/E ACCEPT SMPMCS Modification Control Statements to install HNAS (FMID=LNS0240) HNASXEQ Run HNAS. TSORECV Convert .str files to PDSs
SLNSLOAD	Target library containing the HNAS load module created by APPLY processing.
SLNSMAC	Target macro library (see HNASMAC, above, for additional information)
SLNSSRC	Target source library. Contains source assembled by APPLY processing.
ALNSMAC	Distribution macro library.
ALNSOBJ	Distribution object library (see HNASOBJ, above, for additional information).
ALNSSRC	Distribution source library.

TSO RECEIVE and TRANSMIT Distribution File Information

The TSO TRANSMIT (XMIT) command process generates checksum values which are verified as part of the TSO RECEIVE (RECV) process. This makes these file types more suitable for product distribution than standard binary EBCDIC files. If the RECV operation detects or encounters a corrupt input file, an error message or abnormal termination indication will occur. It is very unusual for file transfer corruption to occur. Should you encounter an error in the RECV process we suggest you review and retry the **binary** transfer process. We refer to these HNAS distribution file types as stream format files in our filename assignment *.STR and respective file descriptions.

Distribution Space Requirement

HNAS non-SMP/E program product libraries typically require the equivalent of 30 cylinders of 3390 space for the average installation. HNAS SMP/E installation requires approximately 4000 tracks on a 3390.

Additional space is required for SYSPRINT files. The output space requirements vary based on environment activity; such as session connect/disconnect frequency, alarm or event filtering (ALRMFLTR=), limits (ALARMLMT=) and TRCPRNT settings.

Distribution (temporary work files) File Disposition

Once the HNAS installation process has completed the distribution stream files (*****.STR) can be deleted:

Non-SMP/E - ***hlq***.HNAS***xxxx***.STR stream files
SMP/E - ***hlq***.LNS0240.F***x***.STR stream files

This page left intentionally blank.

Installation Using Non-SMP/E

This section describes the libraries and procedures used to install the Host Network Access Support (HNAS) program product using the standard Non-SMP/E distribution material.

Distribution File Format and Media Delivery Information

The HNAS product is normally provided in a ZIP file located on the Comm-Pro FTP Server, as an E-mail attachment or on a CD-ROM. The delivery method options are provided by your HNAS sales or support organization.

For non-SMP/e installations the ZIP file name has the following format:

hnas_*vrmnnnn_date_cust#_cid*.zip --> Zip archive file containing non-SMP/E HNAS installation files

Example: hnas_2400045_2004-06-17_99000_cpt.zip

Table 4: HNAS.ZIP File Contents

Filename	Description
hnasgjob.bin	EBCDIC REXX EXEC used to build the hlq.HNASCNTL PDS which contains the jobs used to install and generate HNAS product
hnasmac.str	Stream file created by TSO XMIT for the macro library
hnasmacx.str	Stream file created by TSO XMIT for the macro maintenance library
hnasobj.str	Stream file created by TSO XMIT for the object library
hnasobjx.str	Stream file created by TSO XMIT for the object maintenance library

Installation Overview

The following steps outline the HNAS installation procedure. For these steps, it is assumed that you have selected the appropriate Installation Type (non-SMP/E), acquired the distribution media and located the appropriate distribution source files onto a staging PC in preparation for transfer to the host.

In the following ***hlq*** is the high level data set name qualifier used for all HNAS data sets.

1) Copy the hnasgjob.bin file to a ZOS sequential data set data set (RECFM=FB, LRECL=80, BLKSIZE=3200, 5 tracks will be sufficient). Any data set name may be used. ***tso-user_id***.HNASGJOB.CLIST, where ***tso-user_id*** is your TSO user ID, is suggested. hnasgjob.bin is an EBCDIC REXX EXEC so binary transfer must be used.

2) Edit HNASGJOB to customize it for your installation. Customizing provides the EXEC with your value for ***hlq*** and identifies the DASD volume that will be used for HNAS data sets.

3) Run HNASGJOB EXEC to allocate the ***hlq***.HNASCNTL PDS and load it with members containing the JOBs required to install, run and maintain HNAS. HNASGJOB does not submit any of the jobs it creates.

4) Submit the ***hlq***.HNASCNTL(HNASALOC) job to allocate the data sets used by HNAS.

5) Copy the four .str files to the staging data sets allocated by the HNASALOC JOB (see below for details). The staging data sets are intermediate datasets used by the HNASRCV job to create the HNAS distribution libraries in PDS format.

6) Submit the ***hlq***.HNASCNTL(HNASRCV) JOB to create the PDSs required by HNAS by issuing TSO RECEIVE commands against the .STR data sets.

7) Run the ***hlq***.HNASCNTL(HNASMNT) JOB to assemble HNAS modules with TCP/IP and VTAM interfaces and to linkedit the HNAS load module and store it in ***hlq***.HNASLOAD. The assemblies ensure that HNAS TCP/IP and VTAM interface macros are expanded using IBM macro libraries at your installation. After the HNASMNT completes the next steps are to create a configuration data file (CDF) and an application major node file (AMNF). After these steps, HNAS is ready to be run (see HNASXEQ member in ***hlq***.HNASCNTL). If HNAS is to be installed in an APF authorized library then edit and run the JOB ***hlq***.HNASCNTL(HNASCOPY).

Customizing the HNASGJOB REXX EXEC Parameters

HNASGJOB is a REXX EXEC that allocates the ***hlq***.HNASCNTL data set and loads it with JOBs tailored to your installation's requirements. HNASGJOB must be edited so that the jobs it creates are correct for your installation.

HNASGJOB builds jobs that are stored in ***hlq***.HNASCNTL members. The jobs are not executed until they are submitted by a TSO SUBMIT command. Some of the jobs require further editing. For example the HNASXEQ job needs editing to supply the data set name used for the Configuration Data File (CDF) and the proper job priority. The HNASALOC job (allocate and catalog data sets) may be edited if you wish to change the volumes that certain data sets are placed on. All jobs should be reviewed before they are submitted.

The customizing area is at the start of the EXEC following the REXX comment **/* BEGIN CUSTOMIZATION ...*/**. The statements following the customization comment are changed as follows:

THIS_FILE='*tso-user-id.hnasgjob.clist*'

Replace the string between the quote marks with the fully qualified name of the file you copied hnasgjob.bin to (see Step 1 in the Installation Overview, above). The string shown is the suggested name (where ***tso-user-id*** is your TSO User ID). This step is required because the EXEC contains a data area (actually a large REXX comment) containing prototype statements used to generate the ***hlq***.HNASCNTL members. In order to copy the prototype statements to the target data set the EXEC reads itself. To do this name of the file containing HNASGJOB must be known to HNASGJOB.

Examples:

THIS_FILE='XXX.YYY(HNASGJ)'
hnasgjob.bin was loaded into PDS XXX.YYY as member SMPGJ. The job is run with the TSO command EXEC 'XXX.YYY(HNASGJ)'.

THIS_FILE='SAM.HNASGJOB.CLIST'
hnasjob.bin was loaded into a sequential file. If SAM is the active TSO user id the job is run with the TSO command EXEC HNASGJOB.

JOBCARD1='JJJJXXXX JOB ACNT#,"YYYYYYYYYYYYYY",CLASS=A,'
JOBCARD2='// MSGCLASS=A,NOTIFY=&SYSUID,REGION=0K'

The above two lines are placed at the start of each JOB built by HNASGJOB. JOBCARD1 must end with a ','. The following replacements are made by HNASGJOB:

JJJJ is replaced by the first 4 characters of the active TSO User ID.

XXXX is replaced by descriptive information obtained from the *JOB statement for the job in HNASGJOB (ALOC, XEQ, etc.).

YY..YY is replaced by a 14 character comment obtained from the *JOB statement for the job (e.g. ALOC DATA SETS). The YY..YY string is delimited by 2 single quotes. This allows the replacement string to contain blanks.

Fields other than the replacement fields are copied 'as is' to the generated JOB cards.

Example:

JOBCARD1='HNASXXXX JOB 7777,''YOUR INFO'',CLASS=A,'
causes all JOB names to start with HNAS followed by XXXX replacement data. The YY..YY string has also been removed so 'YOUR INFO' will appear on every JOB statement.

##QL = '*hlq*'
Replace the ***hlq*** string with the high level data set name qualifier that you wish to use. Multiple levels can be used ('ZOS.AAA'). The maximum length for ***hlq*** is 31 characters.

Example:

##QL = 'SYSX.V2R4M0'
indicates the HNAS high level data set name qualifier is SYSX.V2R4M0.

##SYS= '*op-sys-type*'
Replace the ***op-sys-type*** string with:

ZOS
to indicate a ZOS system using MAPI interface to the IBM stack. This is the default value which is correct for most installations. The value OS390 is treated the same as ZOS.

MVS
to indicate that system uses the IUCV interface to communicate with the stack.

Example:

##SYS='ZOS'

##UN = '*uuuu*'
Replace the ***uuuu*** string with the unit type used for HNAS data set allocations.

Example:

##UN = '3390'
indicates a 3390 will be used as the unit type in the JCL generated by the EXEC.

##VL = '*vvvvvv*'
Replace the ***vvvvvv*** string with the volume serial number of the pack used for HNAS data sets. If data set allocation is under SMS control code ##VL='*SMS'. This will cause HNAS to omit VOL=SER= and UNIT= information in the generated JCL. The '*SMS' value is cosmetic -- VOL/SER and UNIT information on a DD statement are ignored if SMS controls data set allocations.

Example:

##VL = 'WORK01'
indicates that WORK01 is the 3390 volume to be used for HNAS data sets.

The following customization statements will need changing if your installation uses non-standard names for IBM components (link editor, assembler) or libraries.

##LK = 'IEWL'
Replace the **IEWL** string if the name of your link editor is not IWL.

##AS = 'ASMA90'
Replace the **ASMA90** string if the name of your assembler is not ASMA90.

##TCM = 'TCPIP.SEZACMAC'
Replace the **TCPIP.SEZACMAC** sting if another name is used for the TCP/IP maclib. This maclib provides macros like EZASMI and IUCV.

##TCL = **'TCPIP.SEZACMTX'**
Replace the **TCPIP.SEZACMTX** string if another name is used for the TCP/IP load library.

##VTM = 'SYS1.MODGEN'
Replace the **SYS1.MODGEN** string if another name is used for the VTAM maclib. This maclib provides macros like NIB and RPL.

##OSM = 'SYS1.MACLIB'
Replace the **SYS1.MACLIB** string if another name is used for the SYSTEM maclib. This maclib provides macros like DCB and WRITE.

The comment **/* END CUSTOMIZATION ... */** delimits the customization area.

Running the HNASGJOB REXX EXEC

If hnasgjob.bin is installed in ***tso-user-id***.HNASGJOB.CLIST then it can be run with the TSO command EXEC HNASGJOB. If installed in XXX.YYY(GJOB) then it can be run with the TSO command EXEC 'XXX.YYY(GJOB)'.

After the HNASGJOB has run, the data set ***hlq***.HNASCNTL has been allocated on the unit and volume specified when the exec was customized. The EXEC also creates the members in ***hlq***.HNASCNTL that will be run to install HNAS (see below).

Running the *hlq*.HNASCNTL Install Jobs

The following describes the jobs in ***hlq***.HNASCNTL in the order that you submit them to install HNAS.

1) HNASALOC

Submit this job to allocate the data sets required by HNAS using the ***hlq,*** unit and volume information specified when the HNASGJOB EXEC was customized. This job must end with a completion code 0.

2) Manual Process - Copy HNAS distribution .str files to the ZOS system

This is a manual operation typically done with IND$FILE or FTP from a local staging PC. The distribution files must be transferred in binary mode. The source and destination file names are as follows:

```
Source (typically a PC)  Z/OS  (Host system)
hnasmac.str  ---------->  hlq.HNASMAC.STR
hnasmacx.str ---------->  hlq.HNASMACX.STR
hnasobj.str  ---------->  hlq.HNASOBJ.STR
hnasobjx.str ---------->  hlq.HNASOBJX.STR
```

The hnas*xxxx*.str files are unloaded PDSs created by the TSO TRANSMIT command. The ***hlq***.HNASxxxx.STR files were allocated by the HNASALOC job. These staging datasets are intermediate datasets that will be used by the HNASRCV job to create the HNAS distribution libraries which are in PDS format. The files were created by TSO XMIT and the staging data sets have an LRECL of 80. IND$FILE transfers must use this LRECL (SEND FILES TO HOST -> OPTIONS -> MVS/TSO -> Transfer Type).

3) HNASRCV

Submit this job to issue TSO RECEIVE commands that create the ***hlq***.HNASMAC, ***hlq***.HNASMACX, ***hlq***.HNASOBJ and ***hlq***.HNASOBJX PDSs from the .STR distribution libraries shown above. After this job has run successfully the ***hlq***.HNAS***xxxx***.STR stream files may be deleted. This job must end with condition code 0.

4) HNASMNT

Submit this job to build the HNAS load module in ***hlq***.HNASLOAD. This job is also run after maintenance has been installed in the hlq.HNASMACX or hlq.HNASOBJX libraries. The job has the following steps (required completion codes in parenthesis):

```
ASM NASMAIN (CC=0)      LINK NASMAIN  (CC=4)
ASM NASTCP  (CC=0)      LINK NASTC    (CC=4)
ASM VTMEXIT (CC=0)      LINK VTMEXIT  (CC=4)
ASM VTMRCV1 (CC=0)      LINK VTMRCV1  (CC=4)
ASM VTMSND1 (CC=0)      LINK VTYMSND1 (CC=4)
ASM VTMSND2 (CC=0)      LINK VTMSND2  (CC=4)
ASM VTMTR   (CC=0)      LINK VTMTR    (CC=4)
ASM VTMUT1  (CC=0)      LINK VTMUT1   (CC=4)
BUILD       (CC=0)
```

The installation process is now complete. Proceed to Configuration Process (CDF and AMNF creation and environment review) section.

A sample copy of the HNASMNT job is at the end of this section.

5) HNASXEQ

This job executes HNAS. The job must be edited to specify the job priority, the name of the CDF to be used and other features. See HNASXEQ member for more information.

Before running the job a configuration data file (CDF) and a VTAM application major node file (AMNF) must be created (see below). HNAS should be run with PARM=FASTRUN (see below) to error check the CDF and to produce the AMNF.

HNASCOPY

This job (PGM=IEBCOPY) may be run to copy the HNAS load module to another library. The job must be edited before it is run.

UNINSTAL

This job (PGM=IDCAMS) may be run at any time to uninstall HNAS and delete all related data sets. The job allows you to start over if errors are made during the installation process. It may also be run in preparation for receiving a refresh HNAS distribution.

Note: This job should be reviewed before it is submitted

Non-SMP/E Maintenance Procedures

Non-SMP/E installations are primarily maintained by refresh (cumulative product maintenance) distributions which replace all HNAS components with components at the current APAR level. The refresh distribution is basically a standard non-SMP/E distribution with maintenance at the latest ***vrmnnnn*** level. APARs are provided with replacement Macro and/or Object members which are placed in the ***hlq***.HNASMAC and ***hlq***.HNASOBJX prior to running the HNASMNT job (see below) . ZAPs (patches) may be provided in emergency cases for pre-APAR problem resolution.

Please refer to the Chapter 5 Maintenance section in the HNAS Guide and Reference manual for a description of the maintenance distribution types and processes.

Sample HNASMNT JOB

```
//HNASMNT  JOB ACNT#,'(RE)BUILD HNAS',CLASS=A,
// MSGCLASS=A,NOTIFY=&SYSUID,REGION=0K
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       JOB TO BUILD (OR REBUILD) HNAS LOAD MODULE
//*
//*       THIS JOB IS RUN AFTER THE INITIAL INSTALLATION OF HNAS FILES
//*       AND AFTER MAINTENANCE HAS BEEN PLACED INTO THE HNASMACX
//*       AND/OR HNASOBJX DATA SETS.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       STEP 1: ASSEMBLE MODULES WITH TCP/IP OR VTAM DEPENDENCIES
//*               USING INSTALLATION'S TCP/IP & VTAM LIBRARIES.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//        JCLLIB ORDER=SYSCPT.@2400000.HNASCNTL
//NASMAIN EXEC NASASM,OBJ=NASMAIN
//ASM.SYSIN DD *
         NASMAIN HOST=ZOS
         END
/*
//NASTCP  EXEC NASASM,OBJ=NASTCP
//ASM.SYSIN DD *
         NASTCP HOST=ZOS
         END
/*
//VTMEXIT EXEC NASASM,OBJ=VTMEXIT
//ASM.SYSIN DD *
         COPY  VTMEXIT
/*
//VTMRCV1 EXEC NASASM,OBJ=VTMRCV1
//ASM.SYSIN DD *
         COPY  VTMRCV1
/*
//VTMSND1 EXEC NASASM,OBJ=VTMSND1
//ASM.SYSIN DD *
         COPY  VTMSND1
/*
//VTMSND2 EXEC NASASM,OBJ=VTMSND2
//ASM.SYSIN DD *
         COPY  VTMSND2
/*
//VTMTR   EXEC NASASM,OBJ=VTMTR
//ASM.SYSIN DD *
         COPY  VTMTR
/*
//VTMUT1  EXEC NASASM,OBJ=VTMUT1
//ASM.SYSIN DD *
         COPY  VTMUT1
/*
//*
```

```
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*      STEP 2:  CREATE THE HOST NAS LOAD MODULE.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BUILD  EXEC  PGM=IEWL,PARM='LIST,LET,SIZE=(384K,96K),AC=1'
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(256,20))
//SYSLIB   DD  DSN=TCPIP.SEZACMTX,DISP=SHR
//SYSLMOD  DD  DSN=SYSCPT.@2400000.HNASLOAD,DISP=OLD
//HNASOBJ  DD  DSN=SYSCPT.@2400000.HNASOBJX,DISP=SHR
//         DD  DSN=SYSCPT.@2400000.HNASOBJ,DISP=SHR
//SYSLIN   DD  DSN=SYSCPT.@2400000.HNASMACX(SYSLIN),DISP=SHR
/*
```

Installation Using SMP/E

This section describes the libraries and procedures used to install the Host Network Access Support (**HNAS**) program product using SMP/E. HNAS may be installed in it's own CSI (Consolidated Software Inventory data set) or in an existing CSI shared by other components. The determination is made by configuration parameters in the SMPGJOB exec (see below).

When HNAS is installed in it's own CSI, DDDEF statements for SMPTLIB are added to the Global Zone. When a shared CSI is used the installation process assumes that all DDDEF entries required for SMP/E execution have already been defined in the appropriate zones. Please refer to your IBM SMP/E documentation guide for information on this process.

The IBM assigned prefix code for HNAS is **LNS**. HNAS at the V2R4M0 level is installed with an **FMID** of LNS0240.

Distribution File Format and Media Delivery Information

The HNAS product is normally provided in a ZIP file located on the Comm-Pro FTP Server, as an E-mail attachment or on a CD-ROM. The delivery method options are provided by your HNAS sales or support organization.

The **lns** file prefix for the *.zip filename indicates a SMP/E product distribution. Please refer to the Product Distribution Media File Information for a description of the *product-level-info* fields.

The ZIP file name has the following format:

lns_*vrmnnnn_date_cust#_cid*.zip   Zip archive file containing SMP/E HNAS installation files.

Example: lns_2400045_2005-06-17_99000_cpt.zip

Table 5: LNS.ZIP File Contents

Filename	Description
smpgjob.bin	EBCDIC REXX EXEC used to generate install JOBs.
lnsjcli.str	Stream file created by TSO XMIT for the JCLIN relfile PDS.
lnsmac.str	Stream file created by TSO XMIT for the macro library relfile PDS.
lnssrc.str	Stream file created by TSO XMIT for the source library relfile PDS.
lnsobj.str	Stream file created by TSO XMIT for the object library relfile PDS.

Installation Overview

SMP/E installation of HNAS requires approximately 4000 tracks on a 3390.

In the following ***hlq*** is the high level data set name qualifier used for all HNAS data sets. ***hlq1*** is the first qualifier in ***hlq***. If ***hlq*** has only one level then ***hlq1*** is identical to ***hlq***. ***rrrr*** is the high level qualifier used for the relative files used by the SMP/E installation process. The customizing process allows the values of ***hlq*** and ***rrrr*** to be specified by you to suit your installation's requirements.

1) Copy the smpgjob.bin file to a ZOS sequential data set data set (RECFM=FB, LRECL=80, BLKSIZE=3200, 5 tracks will be sufficient). Any data set name may be used. ***tso-user_id***.SMPGJOB.CLIST, where ***tso-user_id*** is your TSO user ID, is suggested. smpgjob.bin is an EBCDIC REXX EXEC so binary transfer must be used.

2) Edit SMPGJOB to customize it for your installation. Customizing provides the EXEC with the values of ***hlq*** and ***rrrr*** and identifies the DASD volume that will be used for HNAS data sets. Customizing also provides the name of the existing SMP CSI (if any) to be used.

3) Run SMPGJOB EXEC to allocate the ***hlq***.SLNSCNTL PDS and load it with the JOBs required to install and run HNAS. SMPGJOB does not submit any of the jobs it creates.

4) Submit the ***hlq***.SLNSCNTL(ALOCUCAT and ALOC) jobs to allocate the user catalog and other data sets used by HNAS. These steps may be omitted if you allocate the files by other means.

5) Copy the four .str files to the staging data sets allocated by the ALOC job (see below for details). The staging data sets are intermediate datasets used by the TSORECV job to create the HNAS distribution libraries in PDS format.

6) Submit the ***hlq***.SLNSCNTL(TSORECV) JOB to create the 4 rel file PDSs required for the SMP/E install operation.

7) Run the ***hlq***.SLNSCNTL(SMPRECV, SMPAPLY and SMPACPT) JOBs to install HNAS. After the SMPAPLY or SMPACPT jobs have run ***hlq***.SLNSLOAD contains an HNAS load module. The next steps are to create a configuration data file (CDF) and an application major node file (AMNF). Then HNAS is ready to be run (see member ***hlq***.SLN-SCNTL(HNASXEQ)). If HNAS is to be installed in an APF authorized library then edit and run the JOB ***hlq***.SLNSCNTL(SMPCOPY).

The SMPRECV, SMPAPLY and SMPACPT JOBs all invoke the ***hlq***.SLNSCNTL(SMP-PROC) member. This member is customized for the requirements of the HNAS install. If you use your own SMPPROC be sure that all the data sets in the HNAS SMPPROC SYS-LIB DD list are in your SYSLIB DD list. If this is not done undefined operation codes and other errors will occur when the APPLY step assembles HNAS source members. These assemblies are required because HNAS needs to use your VTAM and TCP/IP communication macros. In addition, your SMPPROC must have DD statements for the HNAS target and distribution libraries.

Customizing the SMPGJOB REXX EXEC Parameters

SMPGJOB is a REXX EXEC that allocates the ***hlq***.SLNSCNTL data set and loads the data set with JOBs tailored to your installation's requirements. SMPGJOB must be edited so that the jobs it creates are correct for your installation.

SMPGJOB builds jobs that are stored in ***hlq***.SLNSCNTL members. The jobs are not executed until they are submitted by a TSO SUBMIT command. Some of the jobs require further editing. For example the HNASXEQ job needs editing to supply the data set name used for the Configuration Data File (CDF) and the proper job priority. The ALOC job (allocate and catalog data sets) may be edited if you wish to change the volumes that certain data sets are placed on. All jobs should be reviewed before they are submitted.

The customizing area is at the start of the EXEC following the REXX comment **/* BEGIN CUSTOMIZATION ...*/**. The statements following the customization comment are changed as follows:

THIS_FILE='*tso-user-id.smpgjob.clist*'

Replace the string between the quote marks with the fully qualified name of the file you copied the smpgjob.bin to (see Step 1 in the Installation Overview, above). The string shown is the suggested name (where ***tso-user-id*** is your TSO User ID). This step is required because the EXEC contains a data area (actually a large REXX comment) containing prototype statements used to generate the ***hlq***.SLNSCNTL members. In order to copy the prototype statements to the target data set the EXEC reads itself. To do this name of the file containing SMPGJOB must be known to SMPGJOB.

Examples:

THIS_FILE='XXX.YYY(SMPGJ)'
smpgjob.bin was loaded into PDS XXX.YYY as member SMPGJ. The job is run with the TSO command EXEC 'XXX.YYY(SMPGJ)'.

THIS_FILE='SAM.SMPGJOB.CLIST'
smpgjob was loaded into a sequential file. If SAM is the active user id the job is run with the TSO command EXEC SMPGJOB.

JOBCARD1='JJJJXXXX JOB ACNT#,"YYYYYYYYYYYYYY",CLASS=A,'
JOBCARD2='// MSGCLASS=A,NOTIFY=&SYSUID,REGION=0K'

The above two lines are placed at the start of each JOB built by SMPGJOB. JOBCARD1 must end with a ','. The following replacements are made by SMPGJOB:

JJJJ is replaced by the first 4 characters of the active TSO User ID.

XXXX is replaced by descriptive information obtained from the *JOB statement for the job in SMPGJOB (RECV, APLY, ACPT, etc.).

YY..YY is replaced by a 14 character comment obtained from the *JOB statement for the job (e.g. ALOC USER CAT). The YY..YY string is delimited by 2 single quotes. This allows the replacement string to contain blanks.

Fields other than the replacement fields are copied ‘as is’ to the generated JOB cards.

Example:

JOBCARD1=’SMPXXXX JOB 7777,’’YOUR INFO’’,CLASS=A,’
causes all JOB names to start with SMP followed by XXXX replacement data. The YY..YY string has also been removed so ‘YOUR INFO’ will appear on every JOB statement.

##CSI = ‘*dsn*’

Replace the ***dsn*** string with the fully qualified name of the existing CSI data set to be used for the installation. Code a null value (‘’) to cause HNAS to allocate it’s own CSI (***hlq***.GLOBAL.CSI). Multiple copies of HNAS may be installed as long as different CSI data sets are used for each copy. We recommend that HNAS have it’s own CSI.

Examples:

##CSI = ‘SYS1.SYSA.GLOBAL.CSI’
Specifies the name of an existing CSI to be used for the install.

##CSI = ‘’
Specifies that HNAS will allocate and initialize ***hlq***.GLOBAL.CSI for the install.

##QL = ‘*hlq*’

Replace the ***hlq*** string with the high level data set name qualifier that you wish to use. Multiple levels can be used (‘ZOS.AAA’). In some cases only the highest level qualifier is used (see second example). ***hlq*** is placed in the DSPREFIX operand in an OPTIONS statement so that the prefix will be used for the SMPTLIB data sets allocated by SMP/E. DSPREFIX has a limit of 26 characters so ***hlq*** may not be longer than 26 characters.

Examples:

##QL = ‘SYSX.V2R4M0’
indicates the HNAS high level data set name qualifier is SYSX.V2R4M0. The user catalog will be named USERCAT.SYSX and SYSX will be an alias in the master catalog connected to USERCAT.SYSX. (only the first level in the qualifier is used for the user catalog name and the alias name). Other data sets will use the full qualifier.

##QL = ‘HNAS240’
indicates the HNAS high level data set name qualifier is HNAS240. The user catalog will be named USERCAT.HNAS240 and HNAS240 will be an alias in the master catalog connected to USERCAT.HNAS240.

##QR = ‘*rrrr*’

Replace the ***rrrr*** string with the high level data set name qualifier that you wish to use for the datasets containing the 4 relative file data set’s required for SMP/E installation. This value is placed in the RFDSNPFX parameter of the ++FUNCTION statement for HNAS. Only a single level qualifier is allowed.

SMP/E does not allow RFDSNPFX to match DSPREFIX (i.e. ***rrrr*** = ***hlq***).

Example:

##QR = ‘SYSX’
The relative files used for the install process will be named as follows:
SYSX.LNS0240.Fi (i=1 to 4). This qualifier does not conflict with
##QL=’SYSX.V2R4M0’. It does conflict with ##QL=’SYSX’

##UN = ‘*uuuu*’

Replace the ***uuuu*** string with the unit type used for HNAS data set allocations.

Example:

##UN = ‘3390’
indicates a 3390 will be used as the unit type in the JCL generated by the EXEC.

##VL = ‘*vvvvvv*’

Replace the ***vvvvvv*** string with the volume serial number of the pack used for HNAS data sets. If SMS allocates data sets the dummy values (which SMS ignores) should be coded for ##VL and ##UNIT.

Example:

##VL = ‘WORK01’
indicates that WORK01 is the 3390 volume to be used for HNAS data sets.

The comment **/* END CUSTOMIZATION ... */** delimits the customization area.

Running the SMPGJOB REXX EXEC

If smpgjob.bin is installed in ***tso-user-id.***SMPGJOB.CLIST then it can be run with the TSO command EXEC SMPGJOB. If installed in XXX.YYY(GJOB) then it can be run with the TSO command EXEC ‘XXX.YYY(GJOB)’.

After the SMPGJOB has run, the data set ***hlq***.SLNSCNTL has been allocated on the unit and volume specified when the exec was customized. The EXEC also creates the members in ***hlq***.SLNSCNTL that will be run to install HNAS (see below).

The exec does not catalog the ***hlq***.SLNSCNTL data set because the user catalog has not been allocated and the alias entry for ***hlq1*** has not been installed in the master catalog. To access the data set the volume serial must be used. This is easily accomplished with the ISPF Data Set List Utility (usually 3.4 off the Primary Option Menu).

Running the *hlq*.SLNSCNTL Install Jobs

The following describes the jobs in ***hlq***.SLNSCNTL in the order that you submit them to install HNAS. All jobs should end with a COND CODE = 0 (there is an exception for SMPACPT, see below).

1) ALOCUCAT

Submit this job to allocate the USERCAT.***hlq1*** user catalog for HNAS. This job also defines ***hlq1*** as an alias in the master catalog. IBM recommends that a separate user catalog be used for products like HNAS. If this job is not run the HNAS data sets will be placed in the master catalog.

2) ALOC

Submit this job to perform the following steps:
Allocate and catalog HNAS and SMP data sets.
Catalog the ***hlq***.SLNSCNTL data set.
Allocate and initialize the HNAS CSI data set (***hlq***.GLOBAL.CSI), if required.
Define the HNAS target and distribution zones in the CSI, define the LNSOPT OPTION so that ***hlq*** is used for SMPTLIB allocations.
Approximately 4000 3390 tracks are required.

3) Manual Process - Copy HNAS distribution .str files to the ZOS system

This is a manual operation typically done with IND$FILE or FTP from a local staging PC. The distribution files must be transferred in binary mode. The source and destination file names are as follows:

```
Source (typically a PC)  Z/OS  (Host system)
lnsjcli.str ---------->  hlq.LNS0240.F1.STR
lnsmac.str  ---------->  hlq.LNS0240.F2.STR
lnssrc.str  ---------->  hlq.LNS0240.F3.STR
lnsobj.str  ---------->  hlq.LNS0240.F4.STR
```

The lns*xxxx*.str files are unloaded PDSs created by the TSO TRANSMIT command. The ***hlq***.LNS0240.Fx.STR files were allocated by the ALOC job. These staging datasets are intermediate datasets that will be used by the TSORECV job to create the HNAS distribution libraries which are in PDS format. The files created by TSO XMIT and the staging data sets have an LRECL of 80. IND$FILE transfers must use this LRECL (SEND FILES TO HOST -> OPTIONS -> MVS/TSO -> Transfer Type).

4) TSORECV

Submit this job to create the following relative file PDSs from the ***hlq***.LNS0240.F*x*.STR files loaded in step 3). The files created (see below) are processed by the SMP/E RECEIVE command. ***rrrr*** (value supplied when SMPGJOB customized) is the DSN prefix which matches the RFDSNPFX() operand on the ++FUNCTION statement (see ***hlq***.SLNSCNTL(SMPMCS)). The LNS0240.F*x* portion of the data set names is required by SMP/E. After this job has run ***hlq***.LNS0240.F*x*.STR files may be deleted.

rrrr.LNS0240.F1
rrrr.LNS0240.F2
rrrr.LNS0240.F3
rrrr.LNS0240.F4

5) SMPRECV

Submit this job to process an SMP/E RECEIVE for the LNS0240 FMID.

6) SMPAPLY

Submit this job to process an SMP/E APPLY for the LNS0240 FMID. The JOB contains the CHECK option to allow checking before data sets are modified. When CHECK is removed the APPLY operation loads the ***hlq***.SLNSMAC and ***hlq***.SLNSSRC target libraries and assembles the HNAS source modules that have dependencies on IBM TCP/IP or IBM VTAM macros. APPLY then link edits the HNAS load module which is placed in ***hlq***.SLNSLOAD target library as member HNAS. HNAS testing can be done at this time. The AUTH DD statement in the HNASXEQ job built by the install process points the NASAUTH member in hlq.SLNSMAC. Jobs and/or PROCs created to start HNAS must have the AUTH DD statement and the NASAUTH member addressed must be the one shipped with the distribution.

If this step ends with a non-zero Condition Code and errors like UNDEFINED OP-CODE then the most likely cause is that the SYSLIB concatenation in SMPPROC does not contain the DD statements required for the HNAS assembly steps. The SMPPROC member in ***hlq*****.SLNSCNTL** has the required libraries (see the SYSLIB DD statement). This problem occurs when a local SMPPROC is used.

7) SMPACPT

Submit this job to process an SMP/E ACCEPT for the LNS0240 FMID. The JOB contains the CHECK option to allow checking before data sets are modified. When CHECK is removed the JOB updates the HNAS ***hlq***.ALNASMAC, ***hlq***.ALSOBJ and ***hlq***.ALNSSRC distribution libraries. The first time the distribution libraries are updated, ACCEPT processing may end with CC=04 and messages with the following form:

GIM24701W   SMP/E COULD NOT OBTAIN LINK-EDIT PARAMETERS.....

This is normal.

8) HNASXEQ

Submit this job to run HNAS. Before running the job a configuration data file (CDF) and a VTAM application major node file (AMNF) must be created (see below). HNAS should be run with the OPTION=FASTRUN (see below) to error check the CDF and produce the AMNF. Additional changes to HNASXEQ are required to specify the job priority and other features. See HNASXEQ member for more information.

9) SMPCOPY

This job (PGM=IEBCOPY) may be run to copy the HNAS load module to another library. The job must be edited before it is run.

UNINSTAL

The UNINSTAL job may be run at any time to un-install HNAS and delete all related data sets. The job allows you to start over if errors are made during the installation process. It may also be run in preparation for receiving a refresh HNAS distribution. When HNAS has it's own CSI the job deletes the entire CSI. When a shared CSI is used the job deletes the HNAS components (zones and the LNSOPT option) that were installed in the shared CSI.
Note: This job should be reviewed before it is submitted

SMP/E Maintenance Procedures

SMP/E installations are maintained by refresh (cumulative product maintenance) distributions which replace all HNAS components with components at the current APAR level. PTFs containing refresh distributions are distributed using the SYSMOD-ID Svrmnnn where vrm is the HNAS version, release & mod level (e.g. 240) and nnn is the highest numbered APAR in the PTF. In unusual circumstances individual APARs may be distributed using the SYSMOD-ID of Tvrmnnn where vrm is the HNAS ver/rel/mod and nnn is the APAR number. If a specific APAR is desired, attention must be paid to the pre- and co-requisites mentioned in the APAR's documentation. If the system is out of date or if the number of changed modules is large, application of a single APAR may require a large number of APARs to be installed. In such cases a refresh PTF is required. Maintenance distributions contain the MCS required to install the SYSMOD. SMP maintenance distributions are provided on request. Please see Chapter 6 (Maintenance Information) for information on installing maintenance with SMP/E.

HNAS may also be upgraded in an SMP/E environment by uninstalling (an UNINSTAL JOB is provided by the installation process) and then installing a new HNAS distribution using the procedures outlined above. This process is often referred to as a HNAS Refresh Reinstall.

Configuration Files

In order for HNAS to operate two configuration files are required.

Configuration Data File (CDF)

The CDF contains statements describing the system configuration to HNAS. This file is created by you based on your original NPSI configuration and your router configuration.

In XOT (Cisco) configurations the statements provide a means of associating TCP/IP sessions (one per active X25 VC) with logical Multi-Channel Links (MCHs) defined by the CDF. This is required because in XOT systems HNAS has no knowledge of the number or location of X25 interfaces on the routers attached to the network. Host stack **TCP port number 1998** is required for **Cisco XOT** router environments.

For XTP (IBM) routers there is one TCP/IP session per router. Configuration statements define routers and the configuration of each X25 MCH link on the routers. In this case the MCHs in HNAS correspond to the MCHs in NPSI. Host stack **TCP port number 3065** is required for **IBM XTP** router environments.

The CDF (reached via the CONFIG DD statement) is processed when HNAS activates. Chapter 3 contains a configuration statement guide and Chapter 4 contains configuration statement reference information. Comm-Pro can generate a “first pass” CDF if you provide us with your NPSI definitions and your Switched Major Node Definitions.

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.

Application Major Node File (AMNF)

The AMNF is a VTAM file containing APPL statements for the VTAM application LUs required for HNAS operation. APPL statements for a CDF are created by HNAS when HNAS is run with the FASTRUN start PARM option specified in conjunction with a MAJNODE DD statement naming the AMNF (see Start Parameter Activation/Run Time Considerations, below). The AMNF must be activated in order for HNAS to operate properly.

<u>We strongly recommend that you use a test or development environment to perform the initial HNAS testing independently of the production environment</u>. We also strongly recommend using the same LU names that NPSI uses for PCNE, PAD and GATE control sessions in order to avoid having to change anything in the application configurations.

Using LU names for HNAS that are different than those used for NPSI may be appropriate for a test environment that is created from scratch specifically for HNAS. For an existing test or production environment, however, we definitely recommend that when replacing NPSI and the 3745 with HNAS and a router, LU names should be maintained to avoid having to also change the host application configurations at the same time. If this recommendation is not followed, a major malfunction can occur which makes it difficult to diagnose problems. For additional information, see “NPSI to HNAS LU Name Migration“ in Chapter 3.

Once a CDF and AMNF have been created, HNAS is ready to be executed. Review the topics in the following sections then proceed to the section titled ‘Starting Host NAS under z/OS, OS/390 and MVS’ later in this chapter.

Router Environment Considerations (Cisco)

The section was created as focal point for Cisco router specific configuration, environment and runtime considerations. Additional content will be added as made available.

HNAS documentation references for Cisco routers follows:

HNAS Guide and Reference Manual - **Chapter 2**, provides **Router Environment Considerations (Cisco)** information (this section).

HNAS Guide and Reference Manual - **Appendix C**, provides Cisco router configuration, debug and display information.

Messages and Codes Debugging Guide - **Cisco Message Associations**, provides Cisco specific or related alert and error message information.

Cisco XOT Configuration Overview

In XOT (Cisco) configurations the statements provide a means of associating TCP/IP sessions (one per active X25 VC) with logical Multi-Channel Links (MCHs) defined by the CDF. This is required because in XOT systems HNAS has no knowledge of the number or location of X25 interfaces on the routers attached to the network. Host stack **TCP port number 1998** is required for **Cisco XOT** router environments.

Cisco Processing of Facilities for Host Initiated Calls (HNAS Callout)

Reference: Cisco's handling of packet size and window size facilities in XOT outbound call request packets.

Problem: Cisco router does not propagate HNAS XOT call request window and packet size facilities when the inbound XOT call request into the router contains like facilities defined on the serial interface x25 link that the call request goes out on.

<u>This condition does not affect the majority of HNAS callout users</u> where (1) the remote network or pad does not require window size and packet size facilities in their inbound call request packets or (2) where the remote x25 resource is configured and uses the window size and packet size of the Cisco serial interface that their x25 link is connected to.

Circumvention: We recommend that you code **'x25 wout *5*'** and **'x25 ops *2048*'** parameters on the respective Cisco interface serial x25 links that support HNAS callout sessions. Choose serial interface supported values that aren't currently in use for your outbound calls so that facilities are properly propagated. We chose these values (***5*** and ***2048***) because they are very unusual and not used in our test environment.

All HNAS host initiated outbound calls (callout sessions) contain facilities (window and packet size) in the XOT call request packet (The XOT protocol requires that these facilities be provided).

We don't understand why the Cisco router does not forward the facilities even though they match their serial interface defaults. Some remote networks or pads require that facilities be provided in their inbound call request packets (some networks reject the call request while other could operate with incorrect values because the remote x25 resource may not receive window and packet size setting that override their defaults that may not match the link level defaults.

You may wish to open a service request ticket with Cisco to see if there is a circumvention for this unusual coding requirement where the Cisco router serial interface outbound window and packet size defaults must be set to a value that isn't the actual default value.

Example 1

In the following example, the router does not forward the HNAS XOT facilities in the call request packet on the serial interface because the values provided in the XOT Call happens to match the routers default serial interface facilities.

cp2621#
dttm.055: [172.29.127.219,1038/10.117.56.100,1998]: **XOT I P/Inactive Call** (37) 8 lci 1
dttm.055: From (6): 912036 To (10): 1036000103
dttm.055: Facilities: **(8)**
dttm.055: Behavior flags:
dttm.055: Reverse charging, no Fast select
dttm.055: Inter-network Call Redirection and Deflection (ICRD) not selected
dttm.059: **Packet sizes: 128 128**
dttm.059: **Window sizes: 2 2**
dttm.059: Call User Data (16): 0x01000000 (pad)
dttm.063: **Serial0/1: X.25 O R1 Call** (31) 8 lci 20
dttm.063: >From (6): 912036 To (10): 1036000103
dttm.063: Facilities: **(2)**
dttm.063: Behavior flags:
dttm.063: Reverse charging, no Fast select
(Packet sizes and window sizes facilities not present, removed by router)
dttm.063: Call User Data (16): 0x01000000 (pad)
dttm.111: Serial0/1: X.25 I R1 Call Confirm (13) 8 lci 20
dttm.111: From (6): 912036 To (10): 1036000103
dttm.111: Facilities: (0)
dttm.115: [172.29.127.219,1038/10.117.56.100,1998]: XOT O P3 Call Confirm (19) 8 lci 1
dttm.115: From (6): 912036 To (10): 1036000103
dttm.115: Facilities: (6)
dttm.115: Packet sizes: 128 128
dttm.115: Window sizes: 2 2

Note: The dttm (datetime field was removed to reduce message length. Only the msec field is present (Month dd hh:mm:ss.msec)

Example 2

In the following example (router interface serial settings **'x25 wout 5'** and **'x25 ops 2048'** - these facilities values are not used in our network environment) you can see that the router forwards the facilities from the xot call request to the serial x25interface call request packet.

cp2621#
dttm.831: [172.29.127.219,1040/10.117.56.100,1998]: **XOT I P/Inactive Call** (37) 8 lci 1
dttm.831: From (6): 912036 To (10): 1036000103
dttm.831: Facilities: **(8)**
dttm.831: Behavior flags:
dttm.831: Reverse charging, no Fast select
dttm.831: Inter-network Call Redirection and Deflection (ICRD) not selected
dttm.831: **Packet sizes: 128 128**
dttm.831: **Window sizes: 2 2**
dttm.835: Call User Data (16): 0x01000000 (pad)
dttm.835: **Serial0/1: X.25 O R1 Call** (37) 8 lci 20
dttm.835: >From (6): 912036 To (10): 1036000103
dttm.839: Facilities: **(8)**
dttm.839: **Packet sizes: 128 128**
dttm.839: **Window sizes: 2 2**
dttm.839: Behavior flags:
dttm.839: Reverse charging, no Fast select
dttm.839: Call User Data (16): 0x01000000 (pad)
dttm.887: Serial0/1: X.25 I R1 Call Confirm (19) 8 lci 20
dttm.887: From (6): 912036 To (10): 1036000103
dttm.891: Facilities: (6)
dttm.891: Packet sizes: 128 128
dttm.891: Window sizes: 2 2
dttm.891: [172.29.127.219,1040/10.117.56.100,1998]: XOT O P3 Call Confirm (19) 8 lci 1
dttm.891: >From (6): 912036 To (10): 1036000103
dttm.891: Facilities: (6)
dttm.895: Packet sizes: 128 128
dttm.895: Window sizes: 2 2

Note: The dttm (datetime field was removed to reduce message length. Only the msec field is present (Month dd hh:mm:ss.msec)

Host Environment Considerations (APF, Security Subsystems, RACF,)

The majority of customer environments can begin testing the HNAS product once the standard installation process is completed and after the HNAS configuration data file (CDF) and router configuration is completed. At some locations we have observed that there are host environment considerations that should be reviewed or resolved before HNAS testing can begin. Please review the following information and address as appropriate:

Authorized Program Facility (APF) Considerations

The HNAS load module **must be stowed** in an **APF registered** library. APF authorization is required so that HNAS can execute in non-swappable mode and allocate its dynamic storage from the high memory area (for example, subpool 230).

The HNAS load module (**HNAS**) is link edited with the **AC=1** option making it a candidate for APF authorization. However, APF authorization is only bestowed when the HNAS load module is placed in a APF registered library. APF registered libraries are identified in the **PROG*xx*** member of your PARMLIB library concatenation (usually SYS1.PARMLIB). The standard HNAS load library (*hlq*.HNASLOAD) or *hlq*.SLNSLOAD(HNAS) can be APF registered by placing its name in the PROG*xx* member.

After the PROG*xx* member has been updated, you will need to issue the following MVS command to put the changes into affect (this will automatically occur at the next operating system IPL).

```
SET PROG=xx              where xx is the last 2 digits in PROGxx
```

HNAS **must execute in non-swappable mode** in order to ensure that data is moved between VTAM and TCPIP in an expeditious manner. To give HNAS the non-swappable property, you will have to add the following statements to the **SCHED*xx*** member of your PARMLIB library concatenation (usually SYS1.PARMLIB).

```
/* HNAS PROPERTIES                                    */
PPT PGMNAME(HNAS)        /* IDENTIFY HNAS LOAD MODULE      */
  CANCEL                 /* ALLOW HNAS TO BE CANCELED      */
  NOSWAP                 /* ALLOW HNAS TO BE NON-SWAPPABLE */
  AFF(NONE)              /* NO PROCESSOR AFFINITY REQUIRED */
```

After the SCHED*xx* member has been updated, you will need to issue the following MVS command to put the changes into affect (this will automatically occur at the next operating system IPL).

```
SET SCH=(xx,L)           where xx is the last 2 digits in SCHEDxx
```

Security Subsystem Registration Considerations

For many installation, **HNAS must be registered** with a resident security subsystem like RACF or Top Secret so that it can use certain IP addresses, processes, datasets, etc.

Failure to register HNAS with the resident security subsystem can prevent HNAS from communicating correctly with the TCPIP stack and in some extreme cases has caused the stack to ABEND. We have observed a condition where the TCPIP stack ABEND's with a 0EC6 cause code when HNAS attempts to assign a socket with the stack. In this case, the stack itself rather than HNAS ceases to function. In some instances, the only way to recover is via a system IPL. While this appears to be a logic error in the TCPIP software security subsystem processing (under investigation by the vendor), it can be avoided by simply registering HNAS as a legitimate TCPIP stack user program.

Normally, if HNAS is not registered with an installation's security subsystem, the TCPIP stack will reject the BIND command that HNAS uses to establish ownership with its server IP address (see the IPADDR= operand of the LOCAL definition statement). When the BIND command fails in this case, HNAS will issue the following alarm message:

NAS2231W SERVER=*ipaddr*(*port*) SOCKID=*xx* PCEID=*yy* NAME=*lclname*
NAS2231W BIND REQUEST FAILED RC=FFFFFFFF 0000000D

The last fullword of the RC= value is the error number. A value of 0000000D (EACCESS) literally means '***permission denied, caller not authorized***'.

There are a number of ways to give HNAS the authority it needs to communicate with the TCPIP stack. Some examples are provided below.

1) One customer that encountered the NAS2231W message describe above used the following RACF commands to give HNAS proper authority.

USERID for HNAS:

```
USER=TCPIP NAME=#STC, FOR TCP/IP
 GROUP=TCPGRP
 ----------------
 UID= 0000000000
 HOME= /
 PROGRAM= /bin/sh
 CPUTIMEMAX= NONE
 ASSIZEMAX= NONE
 FILEPROCMAX= NONE
 PROCUSERMAX= NONE
 THREADSMAX= NONE
 MMAPAREAMAX= NONE
 ***
```

Started-Task for HNAS:

HNAS.* STDATA(USER(TCPIP) TRUSTED(N) PRIVILEGED(N) TRACE(N))

Dataset-Profile for HNAS:

SYP1T0.HNA*.**
TCPGRP ALTER

2) When we encountered the NAS2231W message while running under z/OS V1R4 in our lab, we were able to eliminate it by connecting each user that is allowed to start HNAS to the OMVSGRP as follows:

```
CO userid
   GROUP(OMVSGRP)
   AUTHORITY(USE)
   SPECIAL
   AUDITOR
   UACC(NONE)
```

This works because the OMVS user (and by extension all users who are part of its group) has been given special authority to use system resources. This includes permission in the TCPIP PROFILE to use PORT 1998 for XOT and 3065 for XTP.

RACF Security Subsystem Problem Considerations

Some common problems as well as their circumventions that have be reported while using RACF are listed below:

1) The following error message can be generated if HNAS is registered with RACF but a valid home directory was not specified.

```
ICH408I USER(HNAS) GROUP(ZENT#ADM) NAME(HOST-NETWORK-ACCESS) CL(FSOBJ)
INSUFFICIENT AUTHORITY TO DUB
EFFECTIVE UID(0000010000) EFFECTIVE GID(0000000017)
```

This problem is corrected by adding a home directory (u/OEDFLT) for the default OMVS user.

2) The following error message can be generated if HNAS is registered with RACF but it was not given permission for Read (Search), Write, Directory Access and Execute (it needs all four).

```
ICH408I USER(HNAS) GROUP(ZENT#ADM) NAME(HOST-NETWORK-ACCESS)
/u/OEDFLT CL(DIRSRCH) FID(01C3F0C8C6F0F1000510000000000003)
INSUFFICIENT AUTHORITY TO LOOKUP
ACCESS INTENT(--X) ACCESS ALLOWED(OTHER ---)
EFFECTIVE UID(0000010000) EFFECTIVE GID(0000000017)
```

This problem is corrected by setting the permission bits to 755 for the default OMVS user.

Where: 755 means
- 7.. Read, Write, Directory Access and Execute permission for owner
- .5. Read, Directory Access and Execute permission for group
- ..5 Read, Directory Access and Execute permission for other users

HNAS System Exits

HNAS requires the use of z/OS system exits to process TCPIP and VTAM interrupts. There are a few HNAS modules that are assembled during the HNAS installation process to ensure that they match the level of TCPIP and VTAM you are running. Two of these modules contain the TCPIP and VTAM system exit routines. NASTCP and VTMEXIT, respectively.

The TCPIP system exit routine in NASTCP is called GCSINT. It queues TCPIP interrupt information for task level processing. The equivalent VTAM subroutine in VTMEXIT is called VTMPUSH which is called from other system exit routines in VTMEXIT as well as from VTMRCV1 and VTMSND1. The information passed from these exit routines (interrupt handlers) is processed by task level routines in pre-assembled modules (i.e., modules that are not system release level dependent).

HNAS User Exits

HNAS does not currently provide any user exits. However, if a user requires some customized feature, HNAS can be modified to provide the necessary logic. This is normally done on a time and materials basis.

Host Environment Considerations (z/OS, TCP/IP, VTAM)

z/OS Version/Release Considerations

If you are planning or have installed a new version of z/OS, you ***MUST*** either reinstall HNAS or run the HNAS maintenance installation job (HNASMNT) in order to pick up the most current version of the z/OS TCPIP and VTAM macros. The HNASMNT job is part of the original HNAS distribution (see Chapter 6 Maintenance Information for additional information). Running HNASMNT under the new version of z/OS will make HNAS compatible with the new z/OS version. If HNAS was installed using SMP/E then a refresh PTF must be installed.

TCP/IP Profile Considerations

HNAS (as well as the XOT and XTP protocol) requires a specific TCP port number for the HNAS server component for XOT and XTP router environments. Currently, TCP port number **1998** is **required for Cisco XOT** router environments while TCP port number **3065** is **required for IBM XTP** router environments. Please ensure that these 'well known' TCP port numbers are available for use in your Host stack on the system that you plan to execute HNAS.

Some MVS systems (including Z/OS and OS/390) come pre-configured with limited socket resources defined. The HNAS socket count defaults to a maximum socket count of 2000 unless overridden using the **SOCLMT=*soclmt*** operand from the LOCAL statement section.

HNAS will generate NAS2201W SOCKET FAILED alert messages when the stack rejects its request to open additional socket's (rc-erno). Should you encounter this error we suggest that you adjust the socket count to the number of simultaneous sessions that you plan support under HNAS.

There are parameters in the **TCPIP.PROFILE** file that provided various TCP/IP environment processes and control settings. Some parameters are also located in the **BPXPRM*xx*** member in **SYS1.PARMLIB** (xx=generation level). This member contains parameters that supplement the TCPIP PROFILE file. We suggest that you review parameters **MAXCPUTIME**, **MAXFILEPROC** and **MAXSOCKETS** and increase the resource counts, as required.

MAXSOCKETS for HNAS use must be specified in the INET affinity portion of the BPXPRMxx member. For example:

```
FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)

SUBFILESYSTYPE NAME(TCPIP)
       TYPE(INET)
       ENTRYPOINT(EZBPFINI)

NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
```

```
          MAXSOCKETS(2000)
          TYPE(INET)
          INADDRANYPORT(5555)
          INADDRANYCOUNT(1000)
:
```

MAXCPUTIME and **MAXFILEPROC** are global and are normally specified at the beginning of the BPXPRMxx member. For example:

```
MAXASSIZE(1073741824)
MAXPROCSYS(200)
MAXPROCUSER(100)
MAXUIDS(200)
MAXFILEPROC(2000)
MAXPTYS(256)
MAXTHREADTASKS(5000)
MAXTHREADS(10000)
MAXCPUTIME(86400)
SUPERUSER(OMVSKERN)
CTRACE(CTIBPX00)
:
```

You can change the MAXCPUTIME, MAXFILEPROC and MAXSOCKETS parameters dynamically (until next IPL) using **SETOMVS** commands as follows:

```
SETOMVS MAXCPUTIME=86400         Prevent HNAS SEC6 ABEND
SETOMVS MAXFILEPROC=nnnn         use same value as MAXSOCKETS
SETOMVS MAXSOCKETS=nnnn          HNAS total socket limit
```

Note: The TIME= parameter on the EXEC PGM=HNAS statement is for the operating system. TIME=1440 (24 hours) is magic. It tells the system not to cancel HNAS based on time. TIME=NOLIMIT is also accepted. Contrast this to MAXCPUTIME= which tells the TCPIP stack how long a process (like HNAS) can use it's resources. MAXCPUTIME=86400 tells the stack not to sever it's connection to HNAS based on time. If MAXCPUTIME= is omitted from the BPXPRM*xx* member, a default value is taken from the JCL TIME= operand. In this case, TIME=1440 is treated the same as MAXCPUTIME=86400.

Note: When the ***soclmt*** value is greater than 2000, the following entry for the PORT statement in TCPIP PROFILE file must be added:

1998 TCP *hnasname* NOAUTOLOG SHAREPORT

The **SHAREPORT** option is required when reserving a port to be shared across multiple LISTENers on the same interface. This is true whether the same or different HOME IP addresses are used for each listener.

Note: Regardless of whether SHAREPORT is required, port numbers 1998 (XOT) and 3065 (XTP) must always be reserved for HNAS via the PORT statement.

Host Environment Considerations

As always, please consult with your system administrator before attempting to make any changes to your production/test environment. Additional information on this topic is available under the VCLMT= parameter of the BUILD macro in **Chapter 4** of the product documentation.

Note: For additional TCPIP considerations, please see description of the IPADDR= operand for the LOCAL definition statement in Chapter 4 of the document.

VTAM Considerations

HNAS must execute with the same task dispatching priority as **VTAM** and TCPIP. Specify **CLASS=c, PRTY=14** to set job class ***c*** (VTAM's job class) and the maximum dispatching priority within the job class. **Failure to execute HNAS at the same task dispatching priority as VTAM and TCPIP can cause unexpected results** which can prevent HNAS from communicating correctly with VTAM and the TCPIP Stack. The initial symptoms can be failed transactions and task scheduling problems which can lead to more serious CPU degradation.

Application Considerations

HNAS must execute with the same or higher task dispatching priority (service classes) as the host applications that HNAS is configured to operate with. Failure to do this may result in unexpected sense codes caused by failed operations.

We have observed a condition where a PLU attempt to acquire a HNAS callout resource was rejected by VTAM with the following messages recorded in the system log:

```
IST663I INIT OTHER REQUEST FAILED, SENSE=083A0002 280
IST664I REAL  OLU=USASDV02.NASCTCP    REAL
DLU=USASDV02.TEST16LU
IST889I SID = F93322B8ED1D5647
IST1138I REQUIRED RESOURCE USASDV02.TEST16LU DISABLED
```

The above occurred because the priority of the PLU was higher than the priority of HNAS. HNAS activates a callout LU with two operations: ACB OPEN and SETLOGON OPTCD=START. These operations notify VTAM that the HNAS LU's ACB is open and that the LU will be an SLU. If the PLU attempts to acquire the HNAS resource after the OPEN but before the SETLOGON, the ACQUIRE will fail with the sense shown above.

Please refer to the 'Applications and Products Supported' and 'Application and Vendor Product Notes' sections in Chapter1 of this HNAS Guide and Reference manual for additional information concerning host applications.

Host Environment Considerations (Netview, Sysplex)

Netview Considerations

Console Command Output

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.

Alarm Messages

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.

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.

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, 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 show to the systems operator.

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 PFX WTO 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*).

Note: WTOROUTCDE(ALRM)= support was introduced into 230 under APAR 2300163.

Sysplex Distributor Considerations

HNAS can be used in a Sysplex environment with automatic re acquisition of resources using TCPIP Dynamic Virtual IP Address (DVIPA) support in conjunction with the Sysplex Distributor. The following rules apply:

1) Dynamic VIPAs must be used.

2) The IP address of the Sysplex Distributor must be configured within each HNAS as the IPADDR= operand value of the LOCAL definition statement.

3) The distribution by the Sysplex Distributor of TCPIP resources will be done based on socket IDs and is valid for the duration of the TCPIP session.

4) If one of the participating HNAS is not active, the Sysplex Distributor will recognize this failure and distribute all new sessions to the remaining and active HNAS images.

Activation Process and Run Time Considerations

This section provides information regarding HNAS activation, activation parameter setting and considerations as well as run time considerations.

Starting Host NAS under z/OS, OS/390 and MVS

For z/OS, OS/390 and MVS, HNAS is executed as a start of task. This can be accomplished using a start PROC or running a batch JOB. The sample HNASXEQ Execution JCL that is provided in this section illustrates the JOB set up requirements (This JCL member is also provided as member HNASXEQ in the HNAS *hlq*.HNASCNTL library).

Starting Host NAS for the First Time

Prior to starting HNAS for the first time, it is important that you review the 'Execution and Run Time Considerations' section to understand potential operational issues.

We also strongly recommend that you use a test or development environment to perform the initial HNAS testing independently of the production environment and we advise that you use the same LU names that NPSI uses for PCNE, PAD and GATE control sessions in order to avoid having to change anything in the application configurations.

Using LU names for HNAS that are different than those used for NPSI may be appropriate for a test environment that is created from scratch specifically for HNAS. For an existing test or production environment, however, we definitely recommend that when replacing NPSI and the 3745 with HNAS and a router, LU names should be maintained to avoid having to also change the host application configurations at the same time. If this recommendation is not followed, a major malfunction can occur which makes it difficult to diagnose problems. For additional information, see "NPSI to HNAS LU Name Migration" in Chapter 3.

If you chose to use the same LU names for HNAS and NPSI and NPSI is currently using these LUs then you will need to do the following:

For GATE Fast Connect, inactivate the MCH line and release the PU, inactivate the Fast Connect lines of NPSI and VARY release the associated PUs.

For normal GATE, inactivate the MCH line and VARY release the associated PU and inactivate the associated Switched Major Nodes.

For non GATE LLCs (PCNE, PAD or QLLC) inactivate the NPSI MCH line and inactivate the associated Switched Major Nodes

The HNAS AMNF file should be activated before HNAS is started to prevent inbound sessions from being cleared/rejected. All HNAS LUs are application LUs and must be defined by APPL statements in an Application Major Node File (AMNF).

Note: The **DNAS** console command 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.

Note: The **DMAP APAR** console 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.

Stopping Host NAS under z/OS, OS/390 and MVS

The HNAS program is normally shutdown from the local operator console using various forms of the HNAS **Quit** console command via the modify command interface (for example, **`/F HNASXEQ,QY/`*`conpswd`***). For more information on stopping HNAS, see QUIT command examples in the section titled "Host NAS Console Subsystem Modify Command Interface" on page 85 in this manual).

Additional information on Quit command usage is available under the **QUIT** (terminate operations) section of the "HNAS Console Subsystem Operations Guide".

Note: Some customers prefer to simply Cancel or Purge HNAS from the operator console issuing: **c HNASXEQ** or **p HNASXEQ** We do not recommend stopping HNAS with the (**p**)urge command because all SYSPRINT output will be lost.

Note: The **ALARM LOG=?** console command is now executed 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*.

Execution and Run Time Considerations

HNAS runs with RMODE=24, AMODE=ANY. This is due to the fact that HNAS uses system data management macros which force RMODE=24. This means that HNAS can only execute below the 16M memory boundary. Control blocks, however, can reside anywhere in memory. By convention, all HNAS address pointers are full words. HNAS control blocks are allocated dynamically using system GETMAIN macros based on the information supplied in the Configuration Data File (CDF).

HNAS must execute with the same task dispatching priority as VTAM and TCPIP. Specify **CLASS=c, PRTY=14** to set job class ***c*** (VTAM's job class) and the maximum dispatching priority within the job class. **Failure to execute HNAS at the same task dispatching priority as VTAM and TCPIP can cause unexpected results** which can prevent HNAS from communicating correctly with the TCPIP Stack and VTAM. The initial symptoms can be failed transactions and task scheduling problems which can lead to more serious CPU degradation.

Specify **REGION=0M** to set the **real storage** memory size to the maximum available. The exact memory size (region size) is a function of the HNAS load module size and configuration requirements. You can have HNAS compute the REGION size requirement by first running it with PARM=FASTRUN.

HNAS must not be allowed to terminate based on time expiration. Specify **TIME=1440** or **TIME=NOLIMIT** to prevent HNAS from ABENDing because it's execution time limit has elapsed.

HNAS control storage can be allocated from the **high memory** subpool area. To do this, you must specify a list of subpool numbers in the **APFMEMSP=** suboperand of the PARM= operand. **You should choose subpools from private storage rather than common storage**. Private storage will be released at end of job either explicitly via a FREEMAIN macro when HNAS shuts down gracefully or by the operating system if HNAS is canceled or ABENDs. Common storage is owned by the operating system and can only be release explicitly by a FREEMAIN. If a common storage subpool is specified in the APFMEMSP= suboperand list and HNAS is cancelled, the common storage will be lost until the next operating system IPL. For more information on high memory subpools, refer to the description of the APFMEMSP= suboperand later in this chapter and the IBM MVS Programming: Authorized Assembler Services Guide (GC28-1467).

For sample HNAS start JCL, see HNASXEQ JOB further down in this section.

RACF as well as other security subsystems require that the hlq.HNASLOAD(HNAS) or hlq.SLNSLOAD(HNAS) libraries be registered and that the TSO user name executing the program (via batch job) be registered as well. Please refer to the 'Authorized Program Facility (APF)', 'Security Subsystem Registration' and 'RACF Security Subsystem' Considerations topics under section 'Execution and Run Time Considerations' for additional information.

Host NAS (NASAUTH) Authorization Considerations

Commencing with the V2R2M0 release of HNAS, an authorization file must be provided in order for HNAS to start. The authorization file is identified using the **//AUTH DD** statement in the HNAS start JCL. If the //AUTH DD statement is missing from the start JCL or if it does not point at a valid authorization file, HNAS will not start. Normally, the //AUTH DD statement will point at the **NASAUTH** member in the HNAS MACLIB. The NASAUTH member is an encrypted file containing a **SHIPID** and **expiration date**.

Note: When an HNAS distribution is created, temporary dataset names are remembered by the TSO XMIT command so that they can be reallocated when the TSO RECEIVE command is executed at the customer site. These datasets are allocated during the RECEIVE operation 'under the covers' with no required knowledge of dataset names by the user. For this reason, **the user who is installing HNAS must be authorized to allocate temporary datasets**. If the user does not have proper authorization, temporary dataset allocation will fail and the HNAS distribution will not be installed correctly. If a previous trial version of HNAS is already installed at the customer site, the older NASAUTH file will still be used. This can cause an authorization failure when HNAS is started with the following message being generated:

```
NAS9204S HNAS AUTHORIZATION HAS EXPIRED, NEW KEY REQUIRED
```

For trial users , a generic distribution is provided with a temporary NASAUTH member in the HNAS MACLIB. The temporary NASAUTH member contains either a trial **universal SHIPID** or a **customer specific SHIPID** and a **90-day trial expiration date**. It will allow HNAS to operate for 90 days from the time the distribution was created. This permits customers to conduct tests with HNAS before purchasing the product. During the 90 day trial period, HNAS will report (at midnight) the number of days remaining before authorization expires.

Note: If authorization expires while HNAS is running, it will continue to run. However, if HNAS is stopped, it will not be allowed to restart. In this case, you must obtain a new trail refresh distribution, a permanent distribution, an EOTKEY to extend the trial period or an EOMKEY to dynamically convert the trial distribution to a permanent distribution.

When a trial user accepts and pays for HNAS, **a new permanent (registered) distribution must be installed or an EOMKEY must be applied to dynamically convert the trial distribution to a permanent distribution**. In either case, the trial user is then considered to be a registered user.

For registered users (which includes customers who have purchased older versions of HNAS), **a new registered distribution must be installed** anytime an upgrade or refresh is required in order to pick up the latest maintenance. Comm-Pro no longer provides a generic upgrade or refresh distribution. The registered distribution for each customer will contain a permanent NASAUTH member in the HNAS MACLIB file that is unique to that registered distribution. The permanent NASAUTH member contains a **registered SHIPID** and an **infinite expiration date**. This will allow the registered distribution of HNAS that is on your system to operate indefinitely.

Note: Although a permanent NASAUTH member contains an infinite Trial Period Expiration Date (EOTDATE), it also contains a finite Maintenece/Use Anniversary Date (EOMDATE). A valid EOMDATE must be in effect in order to use HNAS.

Note: The permanent NASAUTH member in a registered distribution cannot be used in any generic distribution or in any other registered distribution. It is valid only for the registered distribution that it was shipped with. Conversely, the temporary NASAUTH member in a generic distribution, cannot be used in any registered distribution but can be used in any other generic distribution unless it has expired.

The DNAS console command can be used to display the SHIPID and distribution creation date.

If the **//AUTH DD** statement is not present in the HNAS start JCL, HNAS will not start. The authorization file is interrogated when HNAS is first started and once per day at midnight, thereafter.

Note: FASTRUN processing is also affected by NASAUTH expiration or mismatches. This is because FASTRUN terminates HNAS execution **after** authorization checking is performed and the DNAS command is executed.

Host NAS Execution DDNAME Requirements

HNAS requires a number of DDNAMEs which are described as follows:

STEPLIB

Points at the load library containing the HNAS load module (HNAS).

SYSABEND

Points at a SYSOUT dataset that can hold a memory dump in the unlikely event of an HNAS ABEND.

Note: We recommend using **SYSABEND** instead of **SYSUDUMP** because SYSABEND will contain everything that SYSUDUMP has plus the LSQA (including subpools 229, 230 and 249) and all IOS control blocks for the HNAS address space.

SYSPRINT

Points at the system spool (SYSOUT=*) or a SYSOUT dataset (=dsn) that will be used initially as the active HNAS log file. We recommend using DCB= parameters RECFM=FBA, LRECL=133 and BLKSIZE=3990 if you do not have other inhouse dsname SYSOUT requirements.

Note: You must set the PRTLMT= operand on the BUILD definition statement to a record count that will not exceed the maximum size of the SYSPRINT dataset or a recoverable ABEND B37 will be encountered which will result in the dataset being closed.

SYSPRNT*x*

Points at alternate SYSOUT datasets that can be used as the active HNAS log file via the **PRNT CLSOPN *ddname*** console command. The SYSPRNT*x* names you specify are arbitrary (new for V2R2M0). The same SYSPRINT PRTLMT=*count* rules apply to these dataset names.

VTAMLIB

Points at the load library containing the LOGTABs and USSTABs that are specified in the HNAS CDF.

MAJNODE

Points at a sequential file that will contain the AMNF that HNAS produces during a CDF FASTRUN pass. This DDNAME is optional. If omitted, HNAS will bypass the AMNF generation portion of the FASTRUN process.

Note: When the FASTRUN process is enable and no MAJNODE DDNAME is provided the HNASXEQ step will encounter a COND CODE 0004 and generate alert message NAS1002W APPLICATION MAJOR NODE FILE COULD NOT BE OPENED, IGNORED.

NEWDEFN

Points at a sequential file that will contain the original CDF and any modifications that are made using the DLCL and DRMT console commands. The DDNAME is only required if the GENNWDF start parameter is specified.

Note: The files for the MAJNODE or NEWDEFN DDNAMES are only used when the FASTRUN or GENNWDF parameters, respectively, are specified. If both FASTRUN and GENNWDF are specified together, FASTRUN wins and GENNWDF is ignored. Therefore, NEWDEFN will only be used if GENNWDF is specified without FASTRUN.

CONFIG

Points at a sequential file containing the HNAS CDF. If omitted, HNAS will terminate.

AUTH

Points at a sequential file containing the HNAS authorization file. If omitted, HNAS will terminate.

Sample HNASXEQ Execution JCL

The following sample JCL can be used to activate HNAS as a batch JOB.

```
//HNASXEQ JOB (),COMMPRO,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//            CLASS=c,PRTY=14,TIME=1440,ADDRSPC=REAL
//*
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*      START HNAS AS A BATCH PROGRAM.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//HNASXEQ EXEC PGM=HNAS,REGION=0M,
//*PARM='RMTCONP,USEMDFY,GENNWDF'                          <- SAMPLE PARMS
// PARM='parmlist'                                         <- REAL PARMS
//STEPLIB  DD DSN=hlq.HNASLOAD,DISP=SHR                    <- APF AUTHORIZED
//SYSABEND DD SYSOUT=*                                     <- REQUIRED
//SYSPRINT DD SYSOUT=*                                     <- REQUIRED
//SYSPRNTn DD SYSOUT=*                                     <- OPTIONAL
//VTAMLIB  DD DSN=SYS1.VTAMLIB,DISP=SHR                    <- LOGTAB/USSTAB
//         DD DSN=hlq.HNASLOAD,DISP=SHR                    <- CUSTOM USSTABS
//AUTH     DD DSN=hlq.HNASMAC(NASAUTH),DISP=SHR            <- REQUIRED
//CONFIG   DD DSN=hlq.HNASMACX(YOURCDF),DISP=SHR           <- REQUIRED
//MAJNODE  DD DSN=hlq.HNASMACX(YOURAMNF),DISP=OLD          <- FASTRUN PARM
//NEWDEFN  DD DSN=hlq.HNASMACX(NEWCDF),DISP=OLD            <- GENNWDF PARM
//cmdlist1 DD DSN=hlq.HNASCNTL(cmslist1),DISP=SHR          <- NEW FOR 240

//
```

Note: For information on SYSPRINT handling, please refer to the PRNT console command description in the Console Subsystem Operations Guide and the section in this Chapter entitled 'SYSPRINT Dataset Output Considerations' on page 2-91.

Host NAS Start Parameters

HNAS accepts a number of start parameters that permit initiation of functions at start of task for various operational processes and environment specific options as well as statistics and debugging options that might otherwise have to be started via a console command. This allows, for example, global PCE statistics to be activated as soon as HNAS is started rather than waiting to be started by a SYSCONS operator.

Note: Most start parameters accept ON or OFF as a follower. An omitted follower is normally treated as ON. If an option is set ON by default (for example, USEMDFY), you must specify OFF as a follower if you wish to disable the option (e.g., USEMDFY OFF).

Note: Some start parameters treat the ON follower or an omitted follower the same as the corresponding console command's ALLON argument. This means that the requested option is global for all associated resources. For example, the TRCDISP [ON] start parameter enables dispatcher tracing for all PCEs just as the TRCDISP ALLON console command does.

Note: Starting with 230 enhancement APAR 2300165, the ALLON and ALLOFF followers will be accepted for the TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCMCH, TRCMCHX and TRCVC start parameters to provide consistency with the console commands of the same name. For start parameters, these followers will be treated the same as ON and OFF, respectively.

Note: Starting with 240 enhancement APAR 2400044, the ALLON and ALLOFF followers will be accepted for the MONTAP start parameter to provide consistency with the MON TAP console command. For the start parameter, these followers will be treated the same as ON and OFF, respectively.

Note: The BUILD CONCMDQ= operand (see Chapter 4) allows console commands to be specified that will be executed (in the order specified) when HNAS starts without operator intervention. This means that you can specify startup functions in addition to or in place of those given in the PARM= operand on the EXEC PGM=HNAS statement.

Starting with 240 Enhancement APAR 2400094, you can now specify all or some of the HNAS start parameters in a sequential file. The syntax for the PARM= operand is as follows: PARM='...,PARMFILE=*pfddname*,...'.

Because of the z/OS JCL limit of 100 characters for the EXEC PARM= operand, you are limited as to how many start parameters you can supply in the PARM= operand. For many HNAS start parameters, there is an equivalent console command which allows you to specify these commands in the EXEC= (or CONCMDQ= operand) on the BUILD definition statement so that the desired functions can be started after the CDF is scanned instead of when HNAS processes it's start parameters. This is a good workaround for the 100 character limit but does not address the problem when a start parameter does not have an equivalent console command.

In order to allow ALL start parameters to be given when HNAS is started, HNAS has been modified to accept a new start parameter: PARMFILE=*pfddname*. *pfddname* identifies a

DDNAME in the HNAS start JCL that represents a sequential file containing a list of other start parameters.

Comments are allowed in the *pfddname* file and must start with an asterisk (*) or semi-colon (;) in record column one (1). Comments can also appear on a parameter record but must start with a semi-colon after the parameter. Each non-comment record in the *pfddname* file has the following format:

parm ; *comment*

Multiple parameters can be specified on a single record as follows:

*parm*1,*parm*2,...,*parm*n ; *comment*

Notes: 1) Leading and trailing blanks are removed from each record before the values are processed.

2) An embedded PARMFILE= parameter in the *pfddname* file will signal the end of data in the file. Any records that follow the embedded PARMFILE= parameter will be ignored and the new *pfddname* file will be processed. An embedded PARMFILE= parameter allows you to chain parameter files. This can also be done by specifying consecutive PARMFILE= parameters in the PARM= operand as follows:

PARM='..,PARMFILE=*pfddname*1,PARMFILE=*pfddname*2,..'

3) When a PARMFILE= parameter is encountered in the PARM= operand, the parameters in the pfddname file are processed before any subsequent parameters in the PARM= operand. After the *pfddname* file is processed, HNAS will process the remaining parameters in the PARM= operand.

The following is an example of the use of 2 parameter files. Assume the HNAS start JCL appears as follows:

```
//LDNAS240 JOB (),COMMPRO,MSGCLASS=X,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*ROUTE XEQ   MVSESA1 //LOADNAS EXEC PGM=HNAS,REGION=0M,
// PARM='APFMEMSP=(230),PARMFILE=PFILE001,PARMFILE=PFILE002,GENNWDF'
//STEPLIB  DD DSN=COMM1.V2R4M0.HNASLOAD,DISP=SHR
//VTAMLIB  DD DSN=SYS1.VTAMLIB,DISP=SHR           <- USSTAB
//*        DD DSN=COMM1.V2R4M0.HNASLOAD,DISP=SHR  <- CUSTOM USSTABS
//*NETVLIB DD DSN=NETVIEW.CNMLINK,DISP=SHR        <- CNMNETV
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//AUTH     DD DSN=COMM1.V2R4M0.HNASMAC(NASAUTH),DISP=SHR
//PFILE001 DD DSN=COMM1.V2R4M0.HNASMAC(PFILE001),DISP=SHR
//PFILE002 DD DSN=COMM1.V2R4M0.HNASMAC(PFILE002),DISP=SHR
//NEWDEFN  DD DSN=COMM1.V2R4M0.NEWCNFG(Z240NEW),DISP=OLD
//CONFIG   DD DSN=COMM1.V2R4M0.OLDCNFG(Z240OLD),DISP=SHR
```

PFILE001 and PFILE002 identify 2 parameter files. These files are listed below:

For PFILE001:

```
* PFILE001
* THIS PARMFILE CONTAINS ADDITIONAL PARMS THAT ARE ADDED
* TO THOSE IN THE EXEC PARM= OPERAND.  SPECIFY AS
* PARM='...,PARMFILE=PFILENAME,...'
*
RMTCONS PRIV , TRCCONS          ; CONSOLE OPTIONS
TRCSUBR                         ; TRACE SUBROUTINE CALLS
TRCWTO                          ; TRACE WTOS
SHOWCMSG                        ; COMPRESS WTOS
```

For PFILE002:

```
* PFILE002
* THIS PARMFILE CONTAINS ADDITIONAL PARMS THAT ARE ADDED
* TO THOSE IN THE EXEC PARM= OPERAND.  SPECIFY AS
* PARM='...,PARMFILE=PFILENAME,...'
*
TRCDBK , TRCBST                 ; TRACE DATA BLOCK AND BINARY
STATS ON                        ; COLLECT STATS
TRCDISP,TRCDATA,TRCBFR,TRCIO    ; TRACE PCE EVENTS
SVRSTRT                         ; ALLOW STACK RESTART
```

After processing the PARM= operand with the 2 parameter files, the result is a PARM= parameter string that is logically equivalent to the following:

```
// PARM='APFMEMSP=(230),RMTCONS PRIV,TRCCONS,TRCSUBR,TRCWTO,SHOWCMSG,TR*
//             CDBK,TRCBST,STATS ON,TRCDISP,TRCDATA,TRCBFR,TRCIO,SVRSTR*
//             T,GENNWDF'
```

The only difference is that the PARM= operand value above could not have been specified in JCL because it exceeds the 100 character limit and thus would have generated a JCL error.

The following start parameters are recognized by the HNAS initiator.

Table 1: Host NAS Start Parameters

PARM	Valid Follower(s)	Description	CC/ VRM
APFXEQ	**{ON|OFF}**	Execution is APF authorized (also sets memory subpool 230 as the default so that APFMEMSP= can be omitted).	**N/220**

Table 1: Host NAS Start Parameters

PARM	Valid Follower(s)	Description	CC/ VRM
APFMEMSP=	(*sp*1,...,*spn*)	Provide memory subpool list (eliminate the need for APFXEQ and the default subpool it establishes).	N/220
DBUG	{TCP}	Bypass certain TCPIP 0198 ABENDs, issuing appropriate alarm messages instead: NAS2109S replaces TCPIP INTERRUPT UNEXPECTED NAS2110S replaces TCPIP REPLY ID FAILURE	A/240 2400102
DUMP	{ON|OFF}	Take dump at end of job.	
EOMKEY=	***dd...dd***	Specifies a 16 decimal digit encoded End of Maintenance/Use Date Extension Key that is used to extend the MAINTENANCE/USE ANNIVERSARY DATE (EOMDATE) without having to install a new HNAS refresh distribution. Note that when an EOMKEY= is used for a trial distriubution, the trial distribution is converted to a permanent distribution without having to install a new HNAS refresh distribution. .	A/240 2400095 A/240 2400106
EOTKEY=	***dd...dd***	Specifies a 16 decimal digit encoded End of Trial Period Date Extension Key that is used to extend the TRIAL PERIOR EXPIRATION DATE (EOTDATE) without having to install a new HNAS refresh distribution.	A/240 2400106
FASTRUN	{ON|OFF}	Process configuration data file for errors, produce AMNF file and memory requirements report then terminate execution.	C/220
GENNWDF	{ON|OFF}	Generate new definition file from the original CDF plus any changes that are made using the MLCL and MRMT console commands.	N/230
MONTAP	{ALLON|ALLOFF} {PKTDATA| MAXDATA| MINDATA| NODATA}	Monitor TAP (Keep Alive) processing in SYSPRINT log for all REMOTEs that have a fixed IP address (not DYNAMIC). Log entries are informational alert messages that use NAS251xM as the message ID. These messages are not written to the SYSCONS (ON or omitted follower also selects the PKTDATA option).	A/230 C/240
PARMFILE=	***pfilename***	Identifies a DDNAME in the HNAS start JCL that points at a sequential file containing other start parameters (see description above).	A/240 2400094

Table 1: Host NAS Start Parameters

PARM	Valid Follower(s)	Description	CC/ VRM
PFXWTO	{ON|OFF|CONS| *text*}	Prefix WTOs with NASNAME= operand value. PFXWTO or PFXWTO ON causes alarm messages to be prefixed. PFXWTO CONS causes console output to be prefixed. To request both options, specify PFXWTO and PFXWTO CONS. PFXWTO *text* is treated as PFXWTO ON but also sets WTO prefix text that overrides the NASNAME= operand value.	C/230
PRNTCNFG	{ON|OFF}	Log configuration information messages (**NAS1*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTDATE	{ON|OFF}	Date stamp SYSPRINT messages.	N/220
PRNTLU	{ON|OFF}	Log LU information messages (**NAS4*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTON		Allow SYSPRINT message logging.	N/230
PRNTQLLC	{ON|OFF}	Log QLLC information messages (**NAS8*xxx*I**) in SYSPRINT when PRNTON is in effect.	A/230
PRNTSYS	{ON|OFF}	Log system information messages (**NAS0*xxx*I** and **NAS9*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTTCP	{ON|OFF}	Log TCP/IP information messages (**NAS2*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTVC	{ON|OFF}	Log VC information messages(**NAS5*xxx*I** and **NASA*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTVTAM	{ON|OFF}	Log VTAM information messages (**NAS3*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTXOT	{ON|OFF}	Log XOT information messages (**NAS7*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
PRNTXTP	{ON|OFF}	Log XTP information messages (**NAS6*xxx*I**) in SYSPRINT when PRNTON is in effect.	N/230
RMTCONP		Allow privileged remote consoles (alias for RMTCONS PRIV).	
RMTCONS	{ON|OFF|PRIV}	Allow remote consoles.	
SHOWCMSG	{ON|OFF}	Display alarm messages in SYSPRINT and on SYSCONS in compressed format (multiple consecutive blanks removed).	A/240 2400036

Table 1: Host NAS Start Parameters

PARM	Valid Follower(s)	Description	CC/ VRM
SHOWCNFG	{**ON**|**OFF**}	Display configuration messages on SYSCONS. The SHOWCNFG option only affects SYSCONS output during HNAS initialization.	
SHOWCONS	{**ON**|**OFF**}	Display console command responses on SYSCONS. The SHOWCONS option has no effect on alarm message output.	
SHOWERR **SHOWOFF** **SHOWON**		Display **error** alarm messages on SYSCONS. The SHOWERR option restricts **informational** alarms to SYSPRINT only. Display **no** alarm messages on SYSCONS. The The SHOWOFF option restricts **all** alarm messages to SYSPRINT only. Display **all** alarm messages on SYSCONS. The SHOWON option allows **all** alarm messages to be displayed on SYSCONS and logged in SYSPRINT. (These thee parameters are mutually exclusive and have no effect on console command output).	
SHOWMORE	{**ON**|**OFF**}	Display additional alarm/trace information on SYSCONS and/or in SYSPRNT.	**A/230**
STATS	{**ON**|**OFF**} {**CONS**|**NETV**|**TCP**| **TMR**|**UTIL**| **XOT**|**XTP**}	Start statistics collection for **all** PCEs (ON or omitted follower also selects all PCE types).	
SVRSTRT	{**ON**|**OFF**}	Restart servers after TCP/IP sever.	
TRCALL	{**ON**|**OFF**}	Start trace for **all** resources (LU, MCH, MCHX, PCE and VC).	
TRCBFR	{**ALLON**|**ALLOFF**} {**CONS**|**NETV**|**TCP**| **TMR**|**UTIL**| **XOT**|**XTP**}	Start buffer trace for **all** PCEs (ON or omitted follower also selects all PCE types).	**C/230**
TRCBFRQ	{**ON**|**OFF**}	Start buffer request trace (XFBFR).	
TRCBST	{**ON**|**OFF**}	Start binary search table trace.	
TRCCNFG	{**ON**|**OFF**}	Start configuration process trace.	**N/220**
TRCCONS	{**ON**|**OFF**}	Start console subsystem trace. **Do not use** unless instructed by Comm-Pro to do so.	**A/230**

Table 1: Host NAS Start Parameters

PARM	Valid Follower(s)	Description	CC/ VRM
TRCDATA	{ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL| XOT|XTP}	Start data trace for **all** PCEs (ON or omitted follower also selects all PCE types).	C/230
TRCDBK	{ON|OFF}	Start data block trace for **all** LUs and VCs .	
TRCDBUG	{ON|OFF}	Start special debug trace. **Do not use** unless instructed by Comm-Pro to do so.	A/230
TRCDISP	{ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL| XOT|XTP}	Start dispatcher trace for **all** PCEs (ON or omitted follower also selects all PCE types).	C/230
TRCIO	{ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL| XOT|XTP}	Start I/O trace for **all** PCEs (ON or omitted follower also selects all PCE types).	C/230
TRCLU	{ALLON|ALLOFF} {MAXDATA| MINDATA| NODATA}	Start trace for **all** LUs (ON or omitted follower also selects the LU MINDATA option).	C/220
TRCMCH	{ALLON|ALLOFF} {ICR|ICRF|OCR| ICLR|OCLR}	Start trace for **all** MCHs (for example, buffer management activity, Call Request failures, etc).	C/240
TRCMCHX	{ALLON|ALLOFF}	Start trace for **all** MCHXs (for example, GATE control session SLU activity).	
TRCPRNT	{ON|OFF}	Log trace records in SYSPRINT.	
TRCSUBR	{ON|OFF} {CONS|MCH|NETV| PCE|TCP|VTAM}	Start subroutine call trace (ON or omitted follower also selects all event types).	C/240
TRCTASK	{ON|OFF}	Start subtask WAIT/POST trace.	
TRCVC	{ALLON|ALLOFF} {MAXDATA| MINDATA| NODATA}	Start trace for **all** VCs (ON or omitted follower also selects the VC MINDATA option).	C/220
TRCWTO	{ON|OFF}	Start WTO alert trace (XFWTO).	
USEMDFY	{ON|OFF}	Use MODIFY rather than WTOR for SYSCONS input.	
USEWTOR	{ON|OFF}	Use WTOR rather than MODIFY for SYSCONS input.	A/230
USENETV	{ON|OFF}	Reserved for future NETVIEW support.	

Note: In the matrix table above, when a command starts an activity for **all** like resources (as TRCMCH ALLON does for **all MCHs**), it means that the action is *global*.

Note: TRCALL ON|OFF is a valid start parameter but rather than being a single trace parameter, it is the logical sum (ORE) of TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCMCH, TRCMCHX, TRCVC, TRCDBK. In other words, it sets all resource traces on or off.

Note: Most start parameters are displayed with a follower like **ON** or **OFF** and most accept a follower as input. **ON** is the implied follower in the absence of one. For example, **USEMDFY** and **USEMDFY ON** are treated identically.

Note: Many start parameters will treat the **ON** or **OFF** followers the same as **ALLON** and **ALLOFF**, respectively, even when ALLON or ALLOFF is displayed as the follower. The commands that support ON|OFF in addition to ALLON|ALLOFF are: TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCMCH, TRCMCHX and TRCVC.

Note: USEMDFY is now a default start parameter. This means that you do not have to code the USEMDFY parameter in the PARM= operand string. Prior to enhancement APAR 2300169, WTOR was the default method for entering console input. With this APAR, the system MODIFY command interface now becomes the default method for entering console input. To restore the WTOR input method, you would have to specify USEMDFY OFF or USEWTOR {ON} as a start parameter. The new USEWTOR parameter (added by APAR 2300169) and the existing USEMDFY parameter are mutually exclusive. USEMDFY ON|OFF is treated the same as USEWTOR OFF|ON, respectively.

Note: The PRNTQLLC start parameter was introduced into 230 under APAR 2300167. Prior to this APAR, NAS8*xxx*I messages were controlled by the PRNTVC start parameter. This new start parameter allows QLLC VC messages to be separated from non-QLLC messages.

The following start parameters require special considerations:

EOTKEY=*dd...dd* allows the TRIAL PERIOD EXPIRATION DATE (**EOTDATE**) for a trial distribution to be extended using a special 16 decimal digit key provided by the ***dd...dd*** digits. An HNAS trial distribution 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 refresh distribution. The new EOTKEY= parameter now allows the EOTDATE to be extended without the need of a new refresh distribution. The EOTKEY=*dd*...*dd* digits are provided by Comm-Pro in a special file that is sent as an e-mail attachment or is downloaded from our FTP server. The EOTKEY=*dd*...*dd* string can then be cut and pasted to the PARM= operand (or MMEM console command). 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
```

The DNAS display for a trial distribution has the following form (for example):

```
HOST NAS INFORMATION FOLLOWS
     HNAS VERSION=V2R4M0 DIST=SMP/E                                    1
     HNAS PROGRAM RUNNING UNDER z/OS 01.11.00                          2
     HNAS PRODUCT INSTALLED UNDER z/OS 01.11.00                        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 2010/11/29                    8
     HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400106        9
     MOST RECENT MAINTENANCE APPLIED IS APAR 2400106                  10
     AUTH=032D SHIPID=1100000011199999                                11
     CUSTID=SFD_99999                                                 12
     CUSTINFO=COMM-PRO ASSOCIATES                                     13
     TRIAL PERIOD EXPIRATION DATE IS 2010/12/31                       14
     DATAFONO SUPPORT IS INCLUDED                                     15
                                                                      16
                                                                      17
     APARID   MAINTENANCE STATUS                                      18
     ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400106              19
```

If EOTKEY=4961000737880526 (for example) is used to extend the EOTDATE, DNAS display lines 11, 14 and 16 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                             16
```

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

EOMKEY=*dd*...*dd* allows the MAINTENANCE/USE ANNIVERSARY DATE (**EOMDATE**) for a permanent distribution to be extended using a special 16 decimal digit key provided by the ***dd*...*dd*** digits. An HNAS permanent distribution 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 EOMKEY= parameter now allows the EOMDATE to be extended without the need of a new refresh distribution. The EOMKEY=*dd*...*dd* digits are provided by Comm-Pro in a special file that is sent as an e-mail attachment or is down-loaded from our FTP server. The EOMKEY=*dd*...*dd* string can then be cut and pasted to the PARM= operand (or MMEM console command). The EOMKEY file has the following format:

```
EOMKEY=4962030747980516
HNAS EOMKEY CREATED AT 16:00:19 ON 2010/11/28
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31
CUSTID=SFD_99999
CUSTINFO=COMM-PRO ASSOCIATES
EMKYDC=0201012311199999
```

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

Note: The EOMKEY= file format was modified via APAR 2400096.

EOMKEY=*dd*...*dd* also allows a trial distribution to be converted 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 16 above as follows:

```
AUTH=000  SHIPID=1100000011199999 EMKYID=1100000011199999      11
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31*                14
EOMKEY=4962030747980516 IS IN EFFECT                           16
```

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.

MONTAP {ALLON|ALLOFF}
{PKTDATA|MAXDATA|MINDATA|NODATA}

This parameter allows you to control TAP monitoring for **all** defined router REMOTEs. It also allows you to specify how much additional information is displayed with each TAP monitor message.

ALLON (or omitted) specifies that TAP monitoring is to be enabled for all router REMOTEs with raw packet data displayed with each TAP monitor message. ALLON is treated the same as if **MONTAP** followed by **MONTAP PKTDATA** were specified. The **PKTDATA** option is assumed when MONTAP is entered by itself or when specified with the ALLON follower.

ALLOFF specifies that TAP monitoring is not to be enabled for any router REMOTE.

PKTDATA (the default) specifies that raw packet data is to be displayed with each TAP monitor message when TAP monitoring is active for any router REMOTE. For example:

```
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
```

MAXDATA specifies that all TAP configuration parameters are to be displayed with each TAP monitor message when TAP monitoring is active for any router REMOTE. For example:

```
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
```

MINDATA specifies that only DTE address parameters are to be displayed with each TAP monitor message when TAP monitoring is active for any router REMOTE. For example:

```
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
```

NODATA specifies that no additional data is to be displayed with each TAP monitor message when TAP monitoring is active for any router REMOTE. For example:

```
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.

TRCALL {ON|OFF}

This parameter allows you to control event tracing for **all** defined resources (all LUs, MCHs, MCHXs, PCEs and VCs).

ON (or omitted) specifies that trace records are to be logged for all LUs, MCHs, MCHXs, PCEs and VCs. **TRCALL [ON]** is treated the same as if start parameters **TRCBFR [ALLON]**, **TRCDATA [ALLON]**, **TRCDBK [ON]**, **TRCDISP [ALLON]**, **TRCIO [ALLON]**, **TRCLU [ALLON]**, **TRCLU MAXDATA**, **TRCMCH [ALLON]**, **TRCMCHX [ALLON]**, **TRCVC [ALLON]** and **TRCVC MAXDATA** were specified.

OFF specifies that trace records are not to be logged for any LU, MCH, MCHX, PCE or VC. Resource traces that are set active by default (e.g., LU, MCH, MCHX and VC) are deactivated. **TRCALL OFF** is treated the same as if start parameters **TRCBFR ALLOFF**, **TRCDATA ALLOFF**, **TRCDBK OFF**, **TRCDISP ALLOFF**, **TRCIO ALLOFF**, **TRCLU OFF**, **TRCLU MINDATA**, **TRCMCH ALLOFF**, **TRCMCHX ALLOFF**, **TRCVC ALLOFF** and **TRCVC MINDATA** were specified.

Note: In 220, **TRCSUBR {ON|OFF}** and **TRCTASK {ON|OFF}** were also controlled by the **TRCALL** command. Starting with 230, **TRCSUBR** and **TRCTASK** parameters are no longer controlled by **TRCALL**. In 230 **TRCSUBR** and **TRCTASK** must be enabled manually as required. This change was made to eliminate unnecessary trace activity.

TRCLU {<u>ALLON</u>|ALLOFF}
{MAXDATA|<u>MINDATA</u>|NODATA}

This parameter allows you to control event tracing for **all** defined LUs. It also allows you to specify how much control block data should be logged with each LU trace entry.

<u>ALLON</u> (or omitted) specifies that trace records are to be logged with the minimum amount of data for all LUs. ALLON is treated the same as if **TRCLU** followed by **TRCLU MINDATA** were specified. The **MINDATA** option is assumed when TRCLU is entered by itself or when specified with the ALLON follower.

ALLOFF specifies that trace records are not to be logged for any LU.

MAXDATA specifies that all control block data (for example, an entire buffer chain) is to be logged with each LU trace entry when tracing is active for any LU.

<u>MINDATA</u> (the default) specifies that some control block data (40 bytes or less) is to be logged with each LU trace entry when tracing is active for any LU.

NODATA specifies that no control block data is to be logged with each LU trace entry when tracing is active for any LU.

Note: The MAXDATA, MINDATA and NODATA followers are used to control how much data should be logged in an LU trace entry. For HNAS release V2R2M0 (prior to APAR 2200047), the MAXDATA, MINDATA and NODATA followers also started event tracing for all defined LUs. This precluded their use to control data logging for a single LU. For this reason, the LU event trace start function has been removed from these followers.

In other words, prior to APAR 2200047, the MAXDATA, MINDATA and NODATA followers started global LU event tracing in addition to controlling the data logging action. This meant that if you wanted to specify the amount of data that was logged for a single LU, you could not do so. After this APAR, these followers no longer start global LU tracing. They simply record the amount of data that is to be logged when and if any LU is traced.

For example, if you want to trace only one LU using MAXDATA, you can issue the following console commands:

```
TRCLU ALLOFF             <- stops global LU tracing if it is active
TRCLU MAXDATA            <- sets MAXDATA option but does not start LU tracing
LUNM=sluname TRCLU ON    <- starts LU tracing for sluname only
```

Note: Once HNAS is running, global LU event tracing must now be started and stopped using the **TRCLU ALLON** and **ALLOFF** console command arguments, respectively

HNAS starts with global LU tracing active and MINDATA set. If you want to start with no LU tracing active but want the MAXDATA option set for subsequent LU tracing via a console command, specify PARM='TRCLU OFF TRCLU MAXDATA'.

TRCMCH {<u>ALLON</u>|ALLOFF}
{ICR|<u>ICRF</u>|OCR|ICLR|OCLR}

This parameter allows you to control event tracing for **all** defined MCHs.

<u>ALLON</u> (or omitted) specifies that trace records are to be logged for all MCHs (also enables ICRF tracing) for all defined MCHs.

ALLOFF specifies that trace records are not to be logged for any MCH (also disables default ICRF tracing) for all defined MCHs.

ICR specifies that all inbound Call Requests are to be traced.

<u>ICRF</u> (the default) specifies that all inbound Call Request failures are to be traced.

OCR specifies that all outbound Call Requests are to be traced.

ICLR specifies that all inbound Clear Requests are to be traced.

OCLR specifies that all outbound Clear Requests are to be traced.

Note: All TRCMCH followers may be entered immediately after one occurrence of the TRC-MCH parameter or via separate TRCMCH invocations. For example, TRCMCH ICR ICLR OCLR is treated the same as TRCMCH ICR,TRCMCH ICLR,TRCMCH OCLR.

TRCMCHX {<u>ALLON</u>|ALLOFF}

This parameter allows you to control event tracing for **all** defined MCHXs.

<u>ALLON</u> (or omitted) specifies that trace records are to be logged for all MCHs.

ALLOFF specifies that trace records are not to be logged for any MCHX.

TRCVC {<u>ALLON</u>|ALLOFF}
{MAXDATA|<u>MINDATA</u>]|NODATA}

This parameter allows you to control event tracing for **all** defined VCs. It also allows you to specify how much control block data should be logged with each VC trace entry.

<u>ALLON</u> (or omitted) specifies that trace records are to be logged with the minimum amount of data for all VCs. ALLON is treated the same as if **TRCVC** followed by **TRCVC MINDATA** were specified. The **MINDATA** option is assumed when TRCVC is entered by itself or when specified with the ALLON follower.

ALLOFF specifies that trace records are not to be logged for any VC.

MAXDATA specifies that all control block data (for example, an entire buffer chain) is to be logged with each VC trace entry when tracing is active for any VC.

MINDATA (the default) specifies that some control block data (40 bytes or less) is to be logged with each VC trace entry when tracing is active for any VC.

NODATA specifies that no control block data is to be logged with each VC trace entry when tracing is active for any VC.

Note: The MAXDATA, MINDATA and NODATA followers are used to control how much data should be logged in a VC trace entry. For HNAS release V2R2M0 (prior to APAR 2200047), the MAXDATA, MINDATA and NODATA followers also started event tracing for all defined VCs. This precluded their use to control data logging for a single VC. For this reason, the VC event trace start function has been removed from these followers.

In other words, prior to APAR 2200047, the MAXDATA, MINDATA and NODATA followers started global VC event tracing in addition to controlling the data logging action. This meant that if you wanted to specify the amount of data that was logged for a single VC, you could not do so. After this APAR, these followers no longer start global VC tracing. They simply record the amount of data that is to be logged when and if any VC is traced.

For example, if you want to trace only those VCs that are active on a specific MCH using MAXDATA, you can issue the following console commands:

```
TRCVC ALLOFF            <- stops global VC tracing if it is active
TRCVC MAXDATA           <- sets MAXDATA option but does not start VC tracing
RNM=mchname TRCVC ON    <- starts VC tracing for mchname only
```

Note: Once HNAS is running, global VC event tracing must now be started and stopped using the **TRCVC ALLON** and **ALLOFF** console command arguments, respectively.

HNAS starts with global VC tracing active and MINDATA set. If you want to start with no VC tracing active but want the MAXDATA option set for subsequent VC tracing via a console command, specify PARM='TRCVC OFF TRCVC MAXDATA'.

TRCBFR|TRCDATA|TRCDISP|TRCIO {ALLON|ALLOFF}
{CONS|NETV|TCP|TMR|UTIL|XOT|XTP}

These parameters allow you to control event tracing for **all** defined PCEs. They also allow you to identify which types of PCEs are to be traced. Note that the followers for all TRC*bddi* parameters are treated identically. For **TRC*bddi*** in the text below, substitute BFR, DATA, DISP or IO for ***bddi***.

If **none** of the PCE type followers (**CONS**, **NETV**, **TCP**, **TMR**, **UTIL**, **XOT** and **XTP**) are specified for a TRC*bddi* parameter, all are assumed. If **any** of the PCE type followers (**CONS**, **NETV**, **TCP**, **TMR**, **UTIL**, **XOT** or **XTP**) are specified for a TRC*bddi* parameter, their functions are ored restricting ***bddi*** tracing to the specified PCE type(s).

<u>ALLON</u> (or omitted) specifies that ***bddi*** trace records are to be logged for all PCE types. ALLON is treated the same as if **TRC*bddi*** followed by **TRC*bddi* CONS NETV TCP TMR UTIL XOT XTP** were specified. All PCE types (**CONS**, **NETV**, **TCP**, **TMR**, **UTIL**, **XOT** and **XTP**) are assumed when TRC*bddi* is entered by itself or when specified with the ALLON follower.

ALLOFF specifies that ***bddi*** trace records are not to be logged for any PCE regardless of its type.

CONS specifies that ***bddi*** trace records are to be logged for all CONSOLE PCEs.

NETV specifies that ***bddi*** trace records are to be logged for all NETVIEW PCEs (NETVIEW PCE support is reserved as a future enhancement).

TCP specifies that ***bddi*** trace records are to be logged for all TCP/IP PCEs.

TMR specifies that ***bddi*** trace records are to be logged for the TIMER PCE.

UTIL specifies that ***bddi*** trace records are to be logged for all UTILITY PCEs.

XTP specifies that ***bddi*** trace records are to be logged for all XTP PCEs.

XOT specifies that ***bddi*** trace records are to be logged for all XOT PCEs.

Note: All TRCBFR, TRCDATA, TRCDSP and TRCIO (TRC*bddi*) followers may be entered immediately after one occurrence of the TRC*bddi* parameter or via separate TRC*bddi* invocations. For example, TRC*bddi* TCP TMR UTIL is treated the same as TRC*bddi* TCP,TRC*bddi* TMR,TRC*bddi* UTIL.

TRCSUBR {ON|OFF} {CONS|MCH|NETV|PCE|TCP|VTAM}

These parameters 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 now 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 now 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.

Host NAS Abbreviated Start Parameters

Abbreviated start parameters are now allowed to reduce the likelihood that the EXEC PARM= operand 100 character limit will be exceeded. For example, **TLU MXDT** can now be specified instead of **TRCLU MAXDATA** which conserves 5 characters of PARM= operand space.

The following table lists the start parameter abbreviations that are supported:

Table 2: Abbreviated Start Parameters

Standard PARM or Follower	Abbreviated PARM or Follower	Standard PARM or Follower	Abbreviated PARM or Follower
ALLOFF	**ALOF**	**ALLON**	**ALON**
APFXEQ	**APFX**	**AUTHCHK**	**ACHK**
CNFGWARN	**CWRN**	**CONCMDQ**	**CNCM**
EOMKEY=	**EK=**	**EOTKEY=**	**TK=**
FASTRUN	**FRUN**	**GENNWDF**	**GNDF**
MAXDATA	**MXDT**	**MINDATA**	**MNDT**
MONTAP	**MTAP**	**NODATA**	**NODT**
PFXWTO	**PWTO**	**PKTDATA**	**PKDT**
PRNTCNFG	**PCFG**	**PRNTDATE**	**PDAT**
PRNTLU	**PLU**	**PRNTOFF**	**POFF**
PRNTON	**PON**	**PRNTQLLC**	**PQLC**
PRNTSYS	**PSYS**	**PRNTTCP**	**PTCP**
PRNTVC	**PVC**	**PRNTVTAM**	**PVTM**
PRNTXOT	**PXOT**	**PRNTXTP**	**PXTP**
RMTCONP	**PCON**	**RMTCONS**	**RCON**
SHOWCMSG	**SCMS**	**SHOWCNFG**	**SCFG**
SHOWCONS	**SCON**	**SHOWERR**	**SERR**
SHOWMORE	**SMOR**	**SHOWOFF**	**SOFF**
SHOWON	**SON**	**SVRSTRT**	**SVRS**
TRCALL	**TALL**	**TRCBFR**	**TBFR**
TRCBFRQ	**TBRQ**	**TRCBST**	**TBST**

Table 2: Abbreviated Start Parameters

TRCCNFG	**TCFG**	**TRCCONS**	**TCON**
TRCDATA	**TDAT**	**TRCDBK**	**TDBK**
TRCDBUG	**TDBG**	**TRCDISP**	**TDSP**
TRCIO	**TIO**	**TRCLU**	**TLU**
TRCMCH	**TMCH**	**TRCMCHX**	**TMCX**
TRCPRNT	**TPRT**	**TRCSUBR**	**TSUB**
TRCTASK	**TTSK**	**TRCVC**	**TVC**
TRCWTO	**TWTO**		

Note: Abbreviated PARM= operand support was introduced into 240 via APAR 2400048.

Host NAS Default Start Parameters

The following table lists the start parameters that are set by default:

Table 3: Default Start Parameters

APFXEQ OFF	DUMP OFF	FASTRUN OFF
GENNWDF OFF	MONTAP OFF	PFXWTO OFF
PRNTCNFG ON	PRNTDATE OFF	PRNTLU ON
PRNTON	PRNTQLLC ON	PRNTSYS ON
PRNTTCP ON	PRNTVC ON	PRNTVTAM ON
PRNTXOT ON	PRNTXTP ON	RMTCONS OFF
SHOWCMSG OFF	SHOWCNFG OFF	SHOWCONS OFF
SHOWERR	SHOWMORE OFF	STATS OFF
SVRSTRT OFF	TRCBFR ALLOFF	TRCBFRQ OFF
TRCBST OFF	TRCCNFG OFF	TRCCONS OFF
TRCDATA ALLOFF	TRCDBK OFF	TRCDBUG OFF
TRCDISP ALLOFF	TRCIO ALLOFF	TRCLU ALLON
TRCLU MINDATA	TRCMCH ALLON	TRCMCH ICRF
TRCMCHX ALLON	TRCPRNT OFF	TRCSUBR OFF
TRCTASK OFF	TRCVC ALLON	TRCVC MINDATA
TRCWTO OFF	USEMDFY ON	USEWTOR OFF
USENETV OFF		

Host NAS Start Parameters versus Console Commands

In most cases, there is an equivalent or similar HNAS console command for each HNAS start parameter. The corresponding console command is generally more robust than the start parameter. Differences are noted in the specific console command description. HNAS console command modifiers (i.e. CID=, ID=, RNM=, etc.) are not supported when using a corresponding start parameter so this further restricts its capabilities.

With the introduction of the **CONCMDQ=** operand of the BUILD definition statement, users are now able to specify complete console command sequences with associated command modifiers for improved trace, SYSCONS and SYSPRINT control. This makes using a start parameter that has an equivalent or similar console command unnecessary. Start parameters that have an equivalent or similar console command are still provided in order to maintain downward compatibly with older releases of HNAS and existing start JCL.

As stated above, the BUILD CONCMDQ= operand (see Chapter 4) allows console commands to be specified that will be executed (in the order specified) when HNAS starts without operator intervention. This means that you can specify startup functions in addition to or in place of those given in the PARM= operand on the EXEC PGM=HNAS statement.

For additional information on HNAS console commands that perform the equivalent or similar functions as HNAS start parameters (except for APFXEQ, APFMEMSP=, DUMP, FASTRUN, GENNWDF, SVRSTRT, USEMDFY and USEWTOR), refer to the HNAS Console Subsystem documentation.

Note: The Operating System's Modify Command Interface (referred to later in this section) is the preferred method used for HNAS Console Subsystem communication. Modify Command processing (**/F *hnasname*,*command***) is requested by coding the **USEMDFY** parameter in the PARM= operand on the EXEC PGM=HNAS statement. If USEMDFY is not specified, access to the Console Subsystem is via the WTOR Interface.

The following table identifies start parameter functions and console command relationships:

Table 4: Host NAS Start Parameters versus Console Commands

PARM	Optional Followers	Equivalent or Similar Console Command	CC/ VRM
APFXEQ	**{ON|OFF}**	N/A	**N/220**
APFMEMSP=		N/A	**N/220**
DUMP	**{ON|OFF}**	QUIT - shutdown option command QA - Quit and Abend option produces a dump.	
EOMKEY=	***dd...dd***	MMEM EOMKEY=*dd...dd*	**A/240**
EOTKEY=	***dd...dd***	MMEM EOTKEY=*dd...dd*	**A/240**
FASTRUN	**{ON|OFF}**	N/A	**C/220**

Table 4: Host NAS Start Parameters versus Console Commands

PARM	Optional Followers	Equivalent or Similar Console Command	CC/ VRM
GENNWDF	**{ON|OFF}**	N/A	**N/230**
MONTAP	**{ALLON|ALLOFF {PKTDATA| MAXDATA| MINDATA| NODATA}**	MON TAP {ALLON|ALLOFF} {PKTDATA|MAXDATA|MINDATA|NODATA}	**A/230** **C/240**
PFXWTO	**{ON|OFF|CONS| *text*}**	PFXWTO **{ON|OFF|CONS|*text*}** (Local Cons) PFXWTO {ON|OFF|TIME} (Remote Cons)	**C/230**
PRNTCNFG	**{ON|OFF}**	PRNT CNFG {ON|OFF}	**N/230**
PRNTDATE	**{ON|OFF}**	PRNT DATE {ON|OFF}	**N/220**
PRNTLU	**{ON|OFF}**	PRNT LU {ON|OFF}	**N/230**
PRNTON		PRNT ON	**N/230**
PRNTQLLC	**{ON|OFF}**	PRNT QLLC {ON|OFF}	**A/230**
PRNTSYS	**{ON|OFF}**	PRNT SYS {ON|OFF}	**N/230**
PRNTTCP	**{ON|OFF}**	PRNT TCP {ON|OFF}	**N/230**
PRNTVC	**{ON|OFF}**	PRNT VC {ON|OFF}	**N/230**
PRNTVTAM	**{ON|OFF}**	PRNT VTAM {ON|OFF}	**N/230**
PRNTXOT	**{ON|OFF}**	PRNT XOT {ON|OFF}	**N/230**
PRNTXTP	**{ON|OFF}**	PRNT XTP {ON|OFF}	**N/230**
RMTCONP		RMTCONS PRIV	
RMTCONS	**{ON|OFF|PRIV}**	RMTCONS {ON|OFF|PRIV}	
SHOWCMSG	**{ON|OFF}**	SHOW CMSG {ON|OFF}	**A/240**
SHOWCNFG	**{ON|OFF}**	N/A	
SHOWCONS	**{ON|OFF}**	SHOW CONS {ON|OFF} The SHOW CONS {ON|OFF} command has no effect on alarm message output.	

Table 4: Host NAS Start Parameters versus Console Commands

PARM	Optional Followers	Equivalent or Similar Console Command	CC/ VRM
SHOWERR		SHOW ERR The SHOW ERR console command restricts **informational** alarms to SYSPRINT only.	
SHOWOFF		SHOW OFF The SHOW OFF console command restricts **all** alarm messages to SYSPRINT only.	
SHOWON		SHOW ON The SHOW ON console command allows **all** alarm messages to be displayed on SYSCONS and logged in SYSPRINT. (These thee commands are mutually exclusive and have no effect on console command output).	
SHOWMORE	**{ON|OFF}**	SHOW {MORE|LESS}	**A/230**
STATS	**{ON|OFF}** **{CONS|NETV|TCP|** **TMR|UTIL|** **XOT|XTP}**	STATS {ON|OFF} {CONS|NETV|TCP| **TMR|UTIL|XOT|XTP}**	
SVRSTRT	**{ON|OFF}**	N/A	
TRCALL	**{ON|OFF}**	TRCALL {ON|OFF}	
TRCBFR	**{ALLON|ALLOFF}** **{CONS|NETV|TCP|** **TMR|UTIL|** **XOT|XTP}**	TRCBFR {ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL|XOT|XTP}	**C/230**
TRCBFRQ	**{ON|OFF}**	TRCBFRQ {ON|OFF}	
TRCBST	**{ON|OFF}**	TRCBST {ON|OFF}	
TRCCNFG	**{ON|OFF}**	TRCCNFG {ON|OFF}	**N/220**
TRCCONS	**{ON|OFF}**	TRCCONS {ON|OFF}	**A/230**
TRCDATA	**{ALLON|ALLOFF}** **{CONS|NETV|TCP|** **TMR|UTIL|** **XOT|XTP}**	TRCDATA {ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL|XOT|XTP}	**C/230**
TRCDBK	**{ON|OFF}**	TRCDBK {ON}|OFF}	
TRCDBUG	**{ON|OFF}**	TRCDBUG {ON}|OFF}	**A/230**
TRCDISP	**{ALLON|ALLOFF}** **{CONS|NETV|TCP|** **TMR|UTIL|** **XOT|XTP}**	TRCDISP {ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL|XOT|XTP}	**C/230**

Table 4: Host NAS Start Parameters versus Console Commands

PARM	Optional Followers	Equivalent or Similar Console Command	CC/ VRM
TRCIO	**{ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL| XOT|XTP}**	TRCIO {ALLON|ALLOFF} {CONS|NETV|TCP| TMR|UTIL|XOT|XTP}	**C/230**
TRCLU	**{ALLON|ALLOFF {MAXDATA| MINDATA| NODATA}**	TRCLU {ALLON|ALLOFF} {MAXDATA|MINDATA|NODATA}	**C/220**
TRCMCH	**{ALLON|ALLOFF} {ICR|ICRF|OCR| ICLR|OCLR}**	TRCMCH {ALLON|ALLOFF} GBL {ICR|ICRF|OCR| **ICLR|OCLR}**	**N/240**
TRCMCHX	**{ALLON|ALLOFF}**	TRCMCHX {ALLON|ALLOFF}	
TRCPRNT	**{ON|OFF}**	TRCPRNT {ON|OFF}	
TRCSUBR	**{ON|OFF} {CONS|MCH|NETV| PCE|TCP|VTAM}**	TRCSUBR {ON|OFF} **{CONS|MCH|NETV| PCE|TCP|VTAM}**	**C/240**
TRCTASK	**{ON|OFF}**	TRCTASK {ON|OFF}	
TRCVC	**{ALLON|ALLOFF} {MAXDATA| MINDATA| NODATA}**	TRCVC {ALLON|ALLOFF} {MAXDATA|MINDATA|NODATA}	**C/220**
TRCWTO	**{ON|OFF}**	TRCWTO {ON|OFF}	
USEMDFY	**{ON|OFF}**	N/A	
USEWTOR	**{ON|OFF}**	N/A	**A/230**
USENETV	**{ON|OFF}**	N/A	

Note: Most start parameters that have an equivalent or similar console command share the same syntax. However, in some cases, a start parameter may be an abbreviation of a console command. For example: The RMTCONP start parameter and the RMTCONS PRIV console command. Abbreviations are sometimes used for start parameters to save space because JCL limits you to 100 characters within the PARM= operand. This is why the CONCMDQ= operand was added to the BUILD definition statement. This operand allows you to specify up to 512 bytes of queued console commands (including a length byte for each command). This makes it possible to automate the tracing function, for example, without having to enter individual commands manually.

Host NAS Start Parameter/Console Command DPARM Considerations

Note: All start parameters can be displayed using the DPARM EXEC console command.

The following table lists those console commands that are equivalent or similar to a start parameter that can toggle the DPARM display output.

Table 5: Host NAS Start Parameter/Console Command DPARM Considerations

Equivalent or Similar Console Command	DPARM EXEC Display Output

`MON TAP {ALLON|ALLOFF}`
`{PKTDATA|MAXDATA|MINDATA|NODATA}`
 	
The MON TAP {ON|OFF} console command is used to control TAP (Keep Alive) monitoring for specific REMOTE(s) and does not effect DPARM output.
	`MONTAP {ALLON|ALLOFF}`
	`MONTAP {PKTDATA|MAXDATA|MINDATA|NODATA}`

`PFXWTO {ON|OFF|CONS|`*text*`}` (Local Cons)	`PFXWTO` *text* `{ON|OFF}`
`PRNT CNFG {ON|OFF}`	`PRNTCNFG {ON|OFF}`
`PRNT DATE {ON|OFF}`	`PRNTDATE {ON|OFF}`
`PRNT LU {ON|OFF}`	`PRNTLU {ON|OFF}`
`PRNT ON`	`PRNTON`
`PRNT QLLC {ON|OFF}`	`PRNTQLLC {ON|OFF}`
`PRNT SYS {ON|OFF}`	`PRNTSYS {ON|OFF}`
`PRNT TCP {ON|OFF}`	`PRNTTCP {ON|OFF}`
`PRNT VC {ON|OFF}`	`PRNTVC {ON|OFF}`
`PRNT VTAM {ON|OFF}`	`PRNTVTAM {ON|OFF}`
`PRNT XOT {ON|OFF}`	`PRNTXOT {ON|OFF}`
`PRNT XTP {ON|OFF}`	`PRNTXTP {ON|OFF}`
`RMTCONS {ON|OFF|PRIV}`	`RMTCONS {ON|OFF|PRIV}`
`SHOW CMSG {ON|OFF}`	`SHOWCMSG {ON|OFF}`
`SHOW CONS {ON|OFF}`	`SHOWCONS {ON|OFF}`
`SHOW {ON|OFF|ERR}`	`SHOWON|SHOWOFF|SHOWERR`
`SHOW {MORE|LESS}`	`SHOWMORE {ON|OFF}`

`STATS {ON|OFF}`
`{CONS|NETV|TCP|`
`TMR|UTIL|XOT|XTP}`
	`STATS {ON|OFF}`
	`STATS {CONS|NETV|TCP|TMR|UTIL|XOT|XTP}`

Table 5: Host NAS Start Parameter/Console Command DPARM Considerations

<table>
<tr><th>Equivalent or Similar Console Command</th><th>DPARM EXEC Display Output</th></tr>
<tr><td>TRCALL ON

The TRCALL ON console command does not alter the TRCLU or TRCVC MAXDATA, MINDATA or NODATA logging option or the TRCMCH ICR, ICRF, OCR, ICLR or OCLR logging option that are currently in effect.</td><td>TRCBFR ALLON
TRCBFR CONS|NETV|TCP|TMR|UTIL|XOT|XTP
TRCDATA ALLON
TRCDATA CONS|NETV|TCP|TMR|UTIL|XOT|XTP
TRCDBK ON
TRCDISP ALLON
TRCDISP CONS|NETV|TCP|TMR|UTIL|XOT|XTP
TRCIO ALLON
TRCIO CONS|NETV|TCP|TMR|UTIL|XOT|XTP
TRCLU ALLON
TRCLU {MAXDATA|MINDATA|NODATA}
TRCMCH ALLON
TRCMCH ICR|ICRF|OCR|ICLR|OCLR
TRCMCHX ALLON
TRCVC ALLON
TRCVC {MAXDATA|MINDATA|NODATA}</td></tr>
<tr><td>TRCALL OFF

The TRCALL OFF console command does not alter the TRCLU or TRCVC MAXDATA, MINDATA or NODATA logging option or the TRCMCH ICR, ICRF, OCR, ICLR or OCLR logging option that are currently in effect</td><td>TRCBFR ALLOFF
TRCBFR NOTYPES
TRCDATA ALLOFF
TRCDATA NOTYPES
TRCDBK OFF
TRCDISP ALLOFF
TRCDISP NOTYPES
TRCIO ALLOFF
TRCIO NOTYPES
TRCLU ALLOFF
TRCLU {MAXDATA|MINDATA|NODATA}
TRCMCH ALLOFF
TRCMCH ICR|ICRF|OCR|ICLR|OCLR
TRCMCHX ALLOFF
TRCVC ALLOFF
TRCVC {MAXDATA|MINDATA|NODATA}</td></tr>
<tr><td>TRCBFR {ALLON|ALLOFF}
{CONS|NETV|TCP|
TMR|UTIL|XOT|XTP}

The TRCBFR {ON|OFF} console command is used to control buffer tracing for specific PCE(s) and does not effect DPARM output.</td><td>TRCBFR {ALLON|ALLOFF}
TRCBFR {CONS|NETV|TCP|
TMR|UTIL|XOT|XTP|NOTYPES}</td></tr>
<tr><td>TRCBFRQ {ON|OFF}</td><td>TRCBFRQ {ON|OFF}</td></tr>
<tr><td>TRCBST {ON|OFF}</td><td>TRCBST {ON|OFF}</td></tr>
<tr><td>TRCCONS {ON|OFF}</td><td>TRCCONS {ON|OFF}</td></tr>
<tr><td>TRCDATA {ALLON|ALLOFF}
{CONS|NETV|TCP|
TMR|UTIL|XOT|XTP}

The TRCDATA {ON|OFF} console command is used to control data tracing for specific PCE(s) and does not effect DPARM output.</td><td>TRCDATA {ALLON|ALLOFF}
TRCDATA {CONS|NETV|TCP|
TMR|UTIL|XOT|XTP|NOTYPES}</td></tr>
</table>

Table 5: Host NAS Start Parameter/Console Command DPARM Considerations

Equivalent or Similar Console Command	DPARM EXEC Display Output
`TRCDBK {[ON}|OFF}`	`TRCDBK {ON|OFF}`

`TRCDISP {ALLON|ALLOFF}`
`{CONS|NETV|TCP|`
`TMR|UTIL|XOT|XTP}`
 	
The TRCDISP {ON|OFF} console command is used to control dispatcher tracing for specific PCE(s) and does not effect DPARM output.
	`TRCDISP {ALLON|ALLOFF}`
	`TRCDISP {CONS|NETV|TCP|`
	`TMR|UTIL|XOT|XTP|NOTYPES}`

`TRCIO {ALLON|ALLOFF}`
`{CONS|NETV|TCP|`
`TMR|UTIL|XOT|XTP}`
 	
The TRCIO {ON|OFF} console command is used to control I/O tracing for specific PCE(s) and does not effect DPARM output.
	`TRCIO {ALLON|ALLOFF}`
	`TRCIO {CONS|NETV|TCP|`
	`TMR|UTIL|XOT|XTP|NOTYPES}`

`TRCLU {ALLON|ALLOFF}`
`{MAXDATA|MINDATA|NODATA}`
 	
The TRCLU {ON|OFF|DBK} console command is used to control tracing for specific LU(s) and does not effect DPARM output.
	`TRCLU {ALLON|ALLOFF}`
	`TRCLU {MAXDATA|MINDATA|NODATA}`

`TRCMCH {ALLON|ALLOFF}`
`GBL {ICR|ICRF|OCR|ICLR|OCLR}`
 	
The TRCMCH {ON|OFF} and TRCMCH {ICR|ICRF|OCR|ICLR|OCLR} (without GBL) console commands are used to control tracing for specific MCH(s) and do not effect DPARM output.
	`TRCMCH {ALLON|ALLOFF}`
	`TRCMCH {ICR|ICRF|OCR|ICLR|OCLR}`

`TRCMCHX {ALLON|ALLOFF}` The TRCMCHX {ON|OFF|DBK} console command is used to control tracing for specific MCHX(s) and does not effect DPARM output.	`TRCMCHX {ALLON|ALLOFF}`
`TRCPRNT {ON|OFF}`	`TRCPRNT {ON|OFF}`
`TRCSUBR {ON|OFF}` **`{CONS|MCH|NETV|`** **`PCE|TCP|VTAM}`**	`TRCSUBR {ON|OFF}` `TRCSUBR` **`{CONS|MCH|NETV|`** **`PCE|TCP|VTAM|NOEVENTS}`**
`TRCTASK {ON|OFF}`	`TRCTASK {ON|OFF}`

`TRCVC {ALLON|ALLOFF}`
`{MAXDATA|MINDATA|NODATA}`
 	
The TRCVC {ON|OFF|DBK} console command is used to control tracing for specific VC(s) and does not effect DPARM output.
	`TRCVC {ALLON|ALLOFF}`
	`TRCVC {MAXDATA|MINDATA|NODATA}`

`TRCWTO {ON|OFF}`	`TRCWTO {ON|OFF}`

Start Parameter Coding Conventions

Start parameters are specified using the PARM= operand on the HNAS EXEC statement. Start parameters must be listed within quotes and separated by commas. You may specify up to 100 characters of information in the PARM= operand which includes the comma separators but excludes the opening and closing quotes. If the list of parameters cannot fit on a single JCL record, continuation records must be specified. All JCL records must start with // in character columns 1 and 2. Data on a record can not exceed character column 71. The continue column is card column 72 which would normally contain a non-blank character to indicate that the next JCL record is a continuation record. However, the continue column need not be marked if a comma is the last character on the record or if the operand being continued is not closed with a quote or right parenthesis. in these cases, a continuation is implied automatically. Continued JCL records for parameters within the PARM= operand must start in character column 16. The first JCL record of the PARM= operand can start as early as character column 4 (e.g., // PARM='...').

As an example, assume the following start parameters are required: USEMDFY, RMTCONP, APFMEMSP=(230,229) (this implies APFXEQ which means that APFXEQ does not have to be specified), TRCDISP, TRCSUBR, TRCLU MAXDATA, TRCVC MAXDATA. To support these parameters, the PARM= operand would be specified as follows:

```
.........1.........2.........3.........4.........5.........6.........7..
123456789012345678901234567890123456789012345678901234567890123456789012
// PARM='USEMDFY,RMTCONP,APFMEMSP=(230,229),TRCDISP,TRCSUBR,TRCLU MAXDA
//             TA,TRCVC MAXDATA'
```

You can also use left and right parenthesis to delimit the PARM= operand list. In this case, any parameter that had a follower would have to be enclosed in quotes (e.g., 'TRCLU MAXDATA') as would any parameter that also used parenthesis as its delimiter's (e.g., 'APFMEMSP=(230,229)'). This would allow you to specify each parameter on a separate JCL record for greater readability, however, the blanks from the end of the last parameter on a record until character column 71 are included in the PARM= string length which is limited to 100 characters. The following example illustrates this coding.

```
.........1.........2.........3.........4.........5.........6.........7..
123456789012345678901234567890123456789012345678901234567890123456789012
// PARM=(USEMDFY,RMTCONP,'APFMEMSP=(230,229)',TRCDISP,TRCSUBR,
//             'TRCLU MAXDATA',
//             'TRCVC MAXDATA')
```

Start Parameter Activation/Run Time Considerations

The **APFXEQ** start parameter may be specified to indicate that HNAS is an **Authorized Program Facility (APF)** registered program in order to allow memory allocation above the 16MB boundary. This mode is enabled by default when **APFMEMSP=** is specified.

The **APFMEMSP=** start parameter may be specified to supply a list of high memory subpools that HNAS will use to allocate its dynamic storage. HNAS can allocate its control blocks above the 16MB boundary to allow substantially larger configurations. This support is enabled when **APFXEQ** or **APFMEMSP=** is coded in the PARM= operand on the HNAS EXEC statement. You may specify up to 7 subpools using the APFMEMSP= suboperand. When APFXEQ is specified and APFMEMSP= is omitted, subpool 230 is assumed. If APFMEMSP= is specified, its list values are processed left to right (APFXEQ is not required when APFMEMSP= is specified). If memory in the first subpool becomes exhausted as control blocks are allocated, the next subpool in the APFMEMSP= list is used. This continues until the end of the list is reached. If all subpools in the list are used before all control blocks are allocated, the low memory area below 16MB is used. If the low memory area still cannot satisfy HNAS memory requirements, HNAS will ABEND. In order for HNAS to use high memory subpools, it is link edited with the **AC=1** option and stowed in an APF registered dataset. The standard HNAS load library (*hlq*.HNASLOAD) can be made APF registered by placing its name in the **LNKLST*xx*** member in **SYS1.PARMLIB**.

The **DUMP** start parameter may be specified to force a memory dump at normal end of job.

The **FASTRUN** start parameter may be specified to allow HNAS to process a Configuration Data File (CDF) without actually initializing. After the CDF is processed, HNAS simply terminates. This option is useful for finding and correcting configuration errors before HNAS is put into service.

Note: The FASTRUN process does not specifically check to see if HNAS is APF authorized. However, this can be verified during a FASTRUN pass by including APFXEQ with the FASTRUN parameter in the PARM= operand on the EXEC PGM=HNAS statement. For example, PARM='FASTRUN,APFXEQ'. If HNAS is not APF authorized, the FASTRUN execution will ABEND with S806. If HNAS is registered with APF, the FASTRUN pass will run to completion.

In addition to checking the CDF, the FASTRUN pass can also create the HNAS AMNF based on all PVC=, SVCi= and LUNAME= operands specified in the CDF. In order to invoke the AMNF generation process, you must specify a **//MAJNODE DD** statement in the HNAS start JOB that points at a sequential file or the member of a PDS with DISP=OLD. The DCB parameters for this AMNF dataset must be RECFM=FB and LRECL=80.

The FASTRUN pass will also produce a memory summary listing that identifies the storage required for all HNAS control blocks as well as the **REGION** size required for run time execution. The control blocks that HNAS requires are identified below in the section titled HNAS Control Blocks.

The **GENNWDF** start parameter may be specified to allow HNAS to generate a new definition file from the original CDF plus any changes that are made while HNAS is executing using the

MLCL and MRMT console commands. All CDF changes are remembered in HNAS memory until HNAS is shutdown. The new CDF is produced at the end of normal HNAS processing. New or modified records are identified in the new CDF by the characters **;NWDF** starting in character position 67.

In order to invoke the new CDF generation process, you must specify a **//NEWDEFN DD** statement in the HNAS start JOB that points at a sequential file or the member of a PDS with DISP=OLD. The DCB parameters for this new CDF dataset must be RECFM=FB and LRECL=80.

The **SVRSTRT** start parameter may be specified to allow HNAS to perform a restart rather than a shutdown if the TCP/IP stack is terminated while HNAS is running. Normally, HNAS will terminate when it detects a TCP/IP stack sever. When this option is specified, HNAS waits for the TCP/IP stack to be restarted. HNAS will also perform this wait if it is started before TCP/IP.

Note: Due to a logic error introduced with multiple server support in February 2002, HNAS does not shutdown when the stack is taken down and the SVRSTRT parameter is omitted. HNAS remains running but does not allow communication with the stack to resume after it is reactivated. In addition when multiple stacks are specified, HNAS will not allow communication with active stacks when any one of them is deactivated. These two problems have been corrected by APAR 2400083. Effective with APAR 2400083, when SVRSTRT is omitted and only one stack is specified in the CDF, HNAS will shutdown when the stack is deactivated. When multiple stacks are specified in the CDF, the SVRSTRT parameter is forced on so that communication with the active stacks will continue even if one is deactivated. In addition, this will allow HNAS to resume communication with a deactivated stack when it is reactivated.

The **SHOWON** start parameter may be specified to **display all** HNAS messages on the system console while the **SHOWOFF** start parameter may be specified to **inhibit all** HNAS messages from being displayed on the system console. The default **SHOWERR** start parameter will **inhibit information only** HNAS messages from being displayed on the system console (only HNAS error messages will be displayed). In all cases, all HNAS messages are written to the HNAS log file (SYSPRINT) unless HNAS BUILD operand **ALRMFLTR= P**urge filters are active.

The **SHOWCMSG** parameter 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:

<u>NAS3799I Without Compression (SHOWCMSG OFF)</u>:

```
0         1         2         3         4         5         6
0123456789012345678901234567890123456789012345678901234567890123456789
```

```
|                                                                     |
NAS3799I LU MCH1PVC  ENDING   SESSION ON MCH MCH1     RMT  CAUSE/DIAG=
000/130 (00/82) DIAGX=0000
|                         |
01234567890123456789012345
7         8         9
```

NAS3799I With Compression (SHOWCMSG ON):

```
0         1         2         3         4         5         6
0123456789012345678901234567890123456789012345678901234567890123456789
|                                                                     |
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.

The **PFXWTO** start parameter may be specified to cause the text provided by the NASNAME= 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.

The **PFXWTO CONS** start parameter may be specified to cause the text provided by the NASNAME= 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 affect. This processing does not affect alarm output.

The **PFXWTO *text*** start parameter may be specified 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.

Specify **PFXWTO *text* PFXWTO OFF PFXWTO CONS** if you only want console output to be prefixed with the *text* value.

Activation and Operation

In 220 and earlier versions, **PRNT** is a default state (not a start parameter) that enables global SYSPRINT logging. The state can be altered using the **PRNT ON** or **PRNT OFF** console command.

In 230 **PRNT** is a start parameter and console command.

Additional information on SYSPRINT control command usage is available under the **PRNT** (SYSPRINT logging control) section of the HNAS **Console Subsystem Operations Guide**.

HNAS tracing and statistics collection require additional computing cycles. These functions can influence HNAS performance and increase system CPU utilization.

Host NAS Console Subsystem Modify Command Interface

The **USEMDFY** start parameter may be specified (or allowed to default) to request the HNAS use the MODIFY command rather than the WTOR reply for Console Subsystem input. **This support is currently available starting with the V1R1M4 release of HNAS**. When the MODIFY interface is selected, the SYSCONS operator or authorized TSO user can enter console commands as follows:

/F *jobname,command*

If you started HNAS using the job on page 2-54 of this document, you can enter the following command to display all SLUs known to HNAS.

/F HNASXEQ,DLU

To display a help list of console Quit options you can enter the following command:

/F HNASXEQ,HELP QUIT

To shutdown HNAS, generate formatted data areas and bypass password prompt, enter:

/F HNASXEQ,QY/*conpswd*

To stop HNAS immediately without formatting data areas, enter:

/F HNASXEQ,QSTOP or /P HNASXEQ

The MODIFY command is useful for automated console access where the knowledge of WTOR reply numbers is generally unavailable.

Please refer to the HNAS **Console Subsystem Operations Guide** for additional information on console operation.

Host NAS Program and Console Output Considerations

HNAS writes messages to SYSCONS (the designated Master/System Operator Console), SYSPRINT and Remote Consoles as enabled and controlled by configuration and start parameters defined by the user. The messages that are generated provide event, alert and operational information for the HNAS environment.

For additional information, see “SYSPRINT Dataset Output Consideration“ in this Chapter, "Output Rules, Default Console Command Output Destination and Default Alarm Message Output Destination" section in the HNAS **Console Subsystem Operations Guide.**

Performance Considerations

This section provides information regarding HNAS performance considerations.

Mainframe

Mainframe CPU usage does increase as NCP 3745 processes (including packet level services) are now running under the host. There are also increased CPU requirements for sessions interfacing with the TCP/IP stack. Generally, if your CPU has available MIPS capacity and is currently loaded below 60%, then adding HNAS to drive some low speed X.25 lines should not be a problem.

Note: HNAS operation will increase the CPU utilization. If your CPU is already highly loaded, an analysis and some relevant tests must be performed to evaluate the impact of the share of the CPU cycles used by HNAS using the Monitor HNAS command that displays the CPU utilization.

HNAS Execution

We suggest that customers reduce unwanted Alarm or Alert message activity using the HNAS CDF ALRMFLTR= and ALRMLMTS= options. Turning off SYSPRINT activity (HNAS Start Parameter TRCPRNT OFF) will further reduce CPU cycle consumption. Customers should also ensure that they haven't left any global traces running (like TRCALL, TRCSUBR or global console command modifiers) which can dramatically affect CPU usage especially when TRCPRNT is enabled.

Note: HNAS alarm/alert event messages, tracing and statistics collection requires additional computing cycles. These functions can influence HNAS performance and increase system CPU utilization.

X.25 Networks

For private X.25 networks, we suggest that customers run with larger packet sizes. Changing from a packet size of 128 to 512 or even higher can dramatically reduce packet processing cycles, especially for file transfer environments. Increasing the packet level window size from 2 to 4 or 7 also improves message processing but no where near the extent of larger packet sizes.

We have measured CPU utilization for file transfers in a representative environment and it was demonstrated that the CPU utilization was divided by 2 when the packet size was set to 512 instead of 128. CPU utilization was divided by 3 when the packet size was set to 1024 in instead of 128. However, increasing the packet size offers no improvement for traffic with short message exchanges. In general, however, even short input requests can generate large amounts of output which can benefit by larger packet sizes.

If public X.25 networks are involved we suggest that you see if the packet size default of 128 can be extended as well as the window-size default of 2. As previously noted, extending the packet-size to a larger size dramatically improves the message processing with reduces CPU requirements.

Performance Considerations

Routers

We suggest that customers disable unnecessary router debug event reporting to reduce router CPU activity when not troubleshooting a problem.

For those employing Cisco access-list Quality of Service (QOS) support, ensure that XOT has the same priority as DLSW and telnet services to avoid choppy screen painting.

ABENDs, Troubleshooting, Problem Determination

When the operating system detects a program exception that is forced (see U0198 NASHALT below) or which occurs due to other factors, an ABEND of HNAS will result. All HNAS ABENDs indicate that a serious problem has occurred. Those that HNAS detects by itself result in a U0198 NASHALT ABEND.

For additional information, please refer to sections entitled System ABEND Codes and HNAS Halt Messages in the HNAS Messages and Codes Debugging Guide.

Contact your HNAS support representative if:

1) you are unable to locate an APAR with corrective logic (PTF) addressing the ABEND condition or

2) you are unable to resolve the problem with the recommendation provided in the document.

When HNAS detects a problem that indicates a severe error has occurred, a NASHALT user 198 ABEND macro is issued to terminate operation and generate a storage dump. Prior to the ABEND the following message is displayed in the job log and in SYSPRINT:

HALT AT LOC *xxxxxxxx* IN *yyyyyyyy*: '*text*'

xxxxxxxx is a storage location where the problem was detected, ***yyyyyyyy*** is the routine name and **'*text*'** is an error message describing the error (for example '**INV LU**').

Once you have located the **HALT AT LOC** and **'*text*'** description or system ABEND code from the SYSCONS or JOBLOG we suggest that you:

1) search for a match in the **HNAS HALT Messages** section of the **HNAS Messages and Codes Guide**,
2) search for a match in the **HNAS APAR Summary** and **Problem Summary** data base files for your particular HNAS edistribution maintenance level.

If you are unable to identify a match or an appropriate resolution with either of these recommendations then please contact your first level HNAS technical support representative for problem determination assistance.

Note: The HNAS edistribution maintenance level can be determined by viewing the output from the HNAS **DNAS** console command display. The display output provides the HNAS VnRnMn edistribution maintenance level, distribution date as well as other valuable information. The display is automatically entered each time HNAS starts (the output is located towards the beginning of the SYSPRINT output) or can be entered from the SYSCONS via the modify interface.

Note: The summary files are located on the HNAS Maintenance web site **www.comm-pro.com/hostnas/maint/index.htm** under 'Host NAS V*n*R*n*M*n* MAINTENANCE (APAR and

PTF) INFORMATION' headings for each supported HNAS release. The summary files are also available on our HNAS FTP server **ftp.comm-pro4ftp.com** under the **hnas_maint** directory for userid registered customers.

Whenever a USER 198 ABEND occurs, it is vital that the SYSPRINT and SYSABEND data sets be prepared for delivery to your HNAS support organization. ABENDs are generally always sent directly to Comm-Pro for analysis. Similarly, in the event of a system ABEND (S0C4, etc.) the SYSPRINT and SYSABEND data sets should also be sent to your HNAS support organization.

In cases where HNAS does not appear to function according to specification, your Level 1 service provider should be contacted for technical support. Problems which cannot be solved by Level 1 are forwarded to Comm-Pro for resolution.

SYSPRINT Dataset Output Consideration

SYSPRINT dataset output considerations for abends, dumps, system logs and host trace output needs to be prepared in a form suitable for processing by a HNAS technical support representative. Our technicians typically load customer provided SYSPRINT output onto their PC's for viewing and analysis. Customers typically provide these files as E-mail attachment or from their HNAS provided FTP userid **/tosup/** directory. The customer SYSPRINT output should be provided as ASCII CRLF delimited files, in EBCDIC RECFM=FB|FBA with standard LRECL= values provided or in EBCDIC RECFM=V variable CRLF delimited format. If you are transferring EBCDIC files we suggest that you generate the files using the TSO Transmit process so that we can receive the files correctly on our z/OS TSO host.

Regardless of the format you choose to send, **please be sure and identify the file formats in a readme.txt file as well as in the E-mail message** providing the problem description for FTP or E-mail file attachment delivery.

Host output files that are preformatted, print image (ASCII or ESCDIC) format that don't require any host programs to format/view will typically be processed by the HNAS technical support representative faster because there are no extra steps involved in locating the files onto our host for processing.

Origin Host System File Types and Format:

1) SYSPRINT - non-Abend, job output,
2) SYSPRINT - formatted dump,
3) Non-SYSPRINT - unformatted dump, IPCS, TRSMAIN (terse), etc.

Provide appropriate DSNAME DSORG, RECFM, LRECL and any other pertinent information concerning the origin file or library.

Shipment file types and format:

1) TSO XMT format,
2) EBCDIC or ASCII sequential file fixed LRECL or CRLF delimited,
3) Compression type if compression program used.

SYSPRINT Output Reduction

Various HNAS parameters and options control the processing, generation and presentation of SYSPRINT and Console output.

In an effort to reduce the amount of SYSPRINT output some non-essential Informational Alert Messages can be purged and not routed to SYSPRINT. Under Purge mode the specific messages are still counted under the ALARM LOG=? message count display area although the individual messages are not written to the Console or SYSRINT.

SYSPRINT Dataset Output Consideration

We recommend adding alert message filter **NAS22**I(P)** to the HNAS BUILD ALRMFLTR= parameter string: Example: ALRMFLTR=(ALLOW,NAS22**I(P)) which will cause HNAS to purge these non-essential TCPIP informational alert messages. While the NAS22**I TCPIP informational alert messages are useful when debugging TCPIP activity they can be ignored during normal HNAS operation. These message types alone generate excessive amounts of SYSPRINT and need only be enabled for viewing when low level TCPIP debugging is required.

Please refer to Chapter4 BUILD ALRMFLTR= parameter section for additional information.

SYSPRINT Dataset and SYSOUT Controls and Operation

SYSPRINT/SYSOUT Note: When the HNASXEQ job (see page 2-53) specifies multiple SYSPRINT datasets, each can be used serially (but not in parallel) via the PRNT CLSOPN *ddname* console command. If you plan to browse 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.

If you want HNAS to automatically switch SYSPRINT datasets, you can use the HNAS 240 (V2R4M0) PRTSWLST operand on the BUILD definition statement.

PRTSWLST=({LOOP|STOP},SWITCHAFTERINIT,SWITCHAT*time*,*ddname*1,...,*ddnamen*) will provide automatic SYSPRINT switching when the current SYSPRINT log file becomes full. The DDNAMEs you specify are used sequentially. The default SYSPRINT file is always used initially (ddname=SYSPRINT).

Note: the ***ddname*i** suboperands of the PRTSWLST can reference static DDNAMEs defined in the HNAS start JCL and/or can request that dynamic SYSPRINT datasets to be used. In the latter case, you would specify **DYNAMIC=*class*** where ***class*** is the JES spool class where the SYSPRINT will be stored. If no class is specified, class A is assumed.

The advantage of using dynamic SYSPRINT files with the **SWITCHAT*time*** action is that a new SYSPRINT file can be allocated everyday to log that day's HNAS activity.   For example, if **PRTSWLST=(LOOP,SWICHATMIDNIGHT,DYNAMIC)** was specified on the BUILD definition statement, a new JES spool file will be allocated for SYSPRINT every day at midnight. Note also that a switch can also occur if the current SYSPRINT file record count reaches the PRTLMT= value regardless of whether a static or dynamic SYSPRINT file is being used.

Please refer to the Chapter4 BUILD PRTSWLST= parameter and the 240 New Features section for additional information.

HNAS Tracing

The HNAS product offers a wide variety of tracing options. Tracing can be restricted by resource type (LU, VC, etc.) or process type (VTAM, TCPIP, etc.). Events can be trapped and tracing suspended when certain packets are sent or received or when specific alarm messages are generated. Trace trapping greatly aids in problem diagnosis since the events leading up to the trap are captured in the frozen HNAS trace table. Please refer to the HNAS Start Parameter sections in Chapter2 of this HNAS Guide and Reference manual for a complete description of trace start parameters (and comparisons with console commands) and the Console Subsystem Guide for a complete description of console trace commands.

General notes on HNAS traces:

In most cases it is best to let HNAS execute with the following default traces for MCH, LU and VC activity (these are system defaults and do not have to be enabled by the users).

TRCLU ALLON
TRCLU MINDATA
TRCMCH ALLON
TRCMCH ICRF
TRCMCHX ALLON
TRCVC ALLON
TRCVC MINDATA

Options like DBK (causes entire LU to be dumped every time the LU is referenced) and TRCSUBR create a lot of output that is seldom required. Default traces run if no trace commands are entered via **/F *hnasname*,*hnas-cmd***, **PARM=(*start parameter*)** operand or the **CONCMDQ=(*hnas-cmd*)**. Note that the PARM= value is located on the batch JOB or Start of Task EXEC statement while the **CONCMDQ=** operand is located on the HNAS CDF BUILD definition statement.

The default traces record MCH, LU and VC activity in an internal trace table. In the event of a severe failure (ABEND) the internal table contains valuable information. There is very little CPU time required to create the internal table entries. The internal table is difficult to read because it has binary information. When the TRCPRNT ON command is entered via **/F** or **PARM=TRCPRNT** then trace records are sent to SYSPRINT in a more readable form. For example, an TSO ‘FIND BIND' command in SYSPRINT will show a BIND command sent by the PLU to an HNAS SLU.

TRCPRNT can generate a lot of output so it should be used with caution. It is possible to trace individual LUs, but for some investigations this is too restrictive should we need to see all LU or VC activity. To stop trace records from going to SYSPRINT all that is required is TRCPRNT OFF (records will only be logged in the internal table).

Recommended trace options for initial session testing:

For troubleshooting <u>SVC Call Request and PVC Setup</u> activity we suggest that you enable **CONCMDQ=(...,'TRCMCH ICR OCR'**) which are invaluable during testing because they show us SVC Call Request and PVC SETUP activity. The resulting trace entries do not normally create large amounts of output.

For troubleshooting <u>SVC Clear Request and PVC Setup Rejection</u> activity we suggest that you enable **CONCMDQ=(...,'TRCMCH ICLR OCLR'**) which are invaluable during testing because they show us SVC Clear Request (and call rejection) activity as well as PVC SETUP rejection activity. The resulting trace entries do not normally create large amounts of output.

If the HNAS product is in test mode or when minimal session activity in underway, we suggest that TRCPRNT be enable via **PARM=TRCPRNT** so that formatted trace entries are written to SYSPRINT for ease of viewing by the user or support representative.

Your HNAS support representative will be happy to work with your organization on a strategy for tracing, trapping and fault isolation. The more information presented to our support representatives the better they will be able to tailor tracing for your specific situation.

HNAS Timer Information and Use

HNAS timers utilize a special one-second HNAS system utility timer that is independent from the time of day clock. Changes to the time of day clock do not affect processing of the HNAS system utility timer process.

System Time of Day Clock

There is one situation where HNAS does utilize the system time of day clock to determine if HNAS was stopped so that a NAS0301E warning message can be issued signifying a 'TIMER LOST INTERRUPT INDICATED' condition. This message may be generated once during a **Daylight Savings** time of day changeover but is not a problem nor will it affect HNAS operation or transaction integrity.

HNAS Timers

The following table represents the HNAS timers. Some values are coded via HNAS CDF parameters while others are fixed.

Table 6: HNAS Timers

Type	Value	Description	
IDLETO=	<u>0</u> 0-255	Specified on the BUILD statement or on an TYPE=MCH|MXT|SPU|XTP REMOTE statement. If an **SVC** (switched virtual circuit) VC session is idle for the number of minutes specified, it is cleared with DIAG=198. IDLETO=0 suppresses inactivity timeout processing.	
MCHTMR=	 <u>60</u> 4-60 nnn	Specified on TYPE=MCH|XTP REMOTE statement in the OPTIONS= parameter. **GATE** - The MCHTMR= option allows you to specify an interval less than one minute which instructs HNAS to monitor the GATE SLUs more often than it normally would. Sub-minute monitoring can reduce or eliminate DIAG=X'85' clear conditions. **QLLC** - The MCHTMR= option triggers HNAS to query all terminal SLUs to ensure that their Application Control Blocks (ACBs) are OPEN so that they can be acquired (accept BIND requests) for callout.	

Table 6: HNAS Timers

Type	Value	Description	
PAUSE	10 1-3600	Suspend or delay execution of **console commands** that follow the PAUSE command for the seconds value specified.	
PVCRECONTMR= PVC Connect to PLU	60 5-254 60	Specified on TYPE=MCH or TYPE=MXT REMOTE statement OPTIONS operand. PVC connect to PLU timer. Causes HNAS to retry connecting a PVC to a PLU after the PVC session has been setup. Connection fails, for example, if the PLU is not active. Prior to Enhancement APAR 2400059, the PVC VTAM connect/reconnect timer was fixed at 60 seconds.	
PVCSETUPTMR= PVC Setup	60 10-254 60 15,120	Specified on TYPE=MCH or TYPE=MXT REMOTE statement OPTIONS operand. PVC Setup request delay (1 minute) after HNAS is initially activated or a REMOTE is activated. Prior to Enhancement APAR 2400059, the PVC SETUP timer was fixed at 60 seconds. No response to HNAS PVC SETUP 15 seconds on first attempt, 2 minutes on subsequent attempts.	
REQSESSDELAY=	2 0-254	Specified on a TYPE=MCH|XTP REMOTE statement in the OPTIONS= parameter to supply a delay (in seconds) between delivery of a **GATE** call request message and the REQSESS macro asking the CTCP for a BIND. REQSESSDELAY=0 implies no delay imposed.	

Table 6: HNAS Timers

Type	Value	Description	
SVCCALLTMR= (Call Request)	30 0-254	Specified on a TYPE=MCH|XTP REMOTE statement in the OPTIONS= parameter specifies how long HNAS should wait for a Call Accepted or Clear Request response to an outbound Call Request packet. Prior to Enhancement APAR 2400069, the Call Request response timer was fixed at 30 seconds.	
TAP=	0 0-4095	Specified on a TYPE=XOT|XTP REMOTE statement. Client router shoulder Tap timeout interval (in seconds). HNAS sends Keep Alive packet to router at the interval specified for TAP=value. Responses must be received within half the TAP= value to avoid **NAS250*ns*** alert message error recovery. TAP=0 suppresses Keep Alive processing (the default for XOT).	
Fixed Timers	-	-	-
Call Request	30	No response to HNAS Call Request: 30 seconds, then clear with DIAG=197. Note: Under **APAR 2400069** this timer can now be specified via OPTIONS= parameter **SVCCALLTMR=*seconds***.	Fixed Timer
Clear Request	10	No response to HNAS Clear: 10 seconds, then close TCP/IP session.	Fixed Timer
REQSESS	10 60	For non PVC sessions: no response to REQSESS sent to VTAM to ask for a BIND from the PLU. The call is cleared with DIAG=143 when the timer expires. For PVC sessions no REQSESS response or completion in error then retry REQSESS in 1 minute.	Fixed Timer
REQUEST SHUTDOWN	20	No response to REQUEST SHUTDOWN sent to PLU: 20 seconds, then send UNBIND to PLU and clear with DIAG=164.	Fixed Timer

Table 6: HNAS Timers

Type	Value	Description	
GATE UNBIND	10	After delivery of a Clear or Clear Confirm packet to a GATE CTCP HNAS waits for the CTCP to end the VTAM session with an UNBIND. If the timeout occurs HNAS closes the VTAM ACB (PLU receives NOTIFY). Session end DIAG=221.	Fixed Timer
QLLC Timers	**-**	**-**	**-**
	10/3	QLLC timers: 10 seconds, retry operation 3 times, then CLEAR with DIAG=89 and the following DIAGX values:	
	-	01 SET MODE TIMEOUT.	
	-	02 DISC TIMEOUT.	
	-	03 XID TIMEOUT.	
	-	04 TEST TIMEOUT.	
	-	05 RESPONSE TIMEOUT.	
	-	09 CLOSED CONNECTION TIMEOUT.	
CLOTINITYP=		Specified on a TYPE=MCH REMOTE statement in the OPTIONS= parameter to supply a Call failure retry count (Clear received or Call Accept response timeout - CLOTFAILRTYLMT={count|3}). Standard Call Accept timeout value is used after Call Request is transmitted.	
TCP/IP Timers	**-**	**-**	**-**
DELAYTIME=	<u>0</u> 0-60	Specified on a TYPE=XOT|XTP LOCAL statement in the INIT= parameter to supply a delay (in seconds) after a TCP/IP BIND failure before another BIND is attempted to connect a server socket (IPADDR/PORT) to the stack. DELAYTIME=0 inhibits any delay.	
GETIBMOPT	60	Fixed timer (in seconds) between successive GETIBMOPT requests that indicate TCPIP stack is down or not defined. GETIBMOPT is used to interrogate the named stack (from the TCPNAME= operand) before an INITAPI is issued to connect HNAS to the stack.	

HNAS Timers

Table 6: HNAS Timers

Type	Value	Description	
INITAPI	60	Fixed timer (in seconds) between successive INITAPI requests that indicate TCPIP stack is down or not defined. INITAPI is used after a successful GETIBMOPT request to connect HNAS to the stack named by the TCPNAME= operand.	
SELECT	60	Fixed timer (in seconds) that is started when a SELECT request is issued. SELECT normally ends an error is detected, when input has arrived or when a stack requested timeout occurs (forced end when no errors or input is pending). If the SELECT does not end normally (as the result of one of the conditions just mentioned) the HNAS SELECT timeout will CANCEL the SELECT and then reissue it. This timer is a work around for an HNAS hang that we feel is a stack problem (PMRs 82217 and 83755 have been opened with IBM for a resolution).	
Console Timers	**-**	**-**	**-**
PING (XOT Call Request response)	15	No response to HNAS PING request.	**Fixed Timer**

X25 Level 3 Timers

The following table represents some of the common X25 Level 3 timers and default values. Some values may be coded via HNAS CDF parameters or Cisco router (interface serial x25 tn value) parameters while others are fixed.

Table 7: X25 Level 3 Timers

Type	Timer Description	HNAS	CISCO
t10	Restart Indication (DCE) retransmission timer	n/a	60
t11	Incoming Call timer (DCE)	n/a	180
t12	Reset Indication (DCE) retransmission timer	n/a	60
t13	Clear Indication (DCE) retransmission timer	n/a	60
t20	Restart Request retransmission timer	n/a	180
t21	Call Request timer HNAS OPTIONS=SVSCALLTMR= (2400069) (HNAS closes socket after timeout)	30 10-254	200
t22	Reset Request retransmission timer	0	180
t23	Clear Request retransmission timer (HNAS closes socket after timeout)	10	180

Host NAS control blocks

One Process Control Element (**PCE**) is allocated for each TCP/IP socket that HNAS uses. One TCP/IP socket is required for each XTP or XOT server and client component connection. A server component is defined by the LOCAL definition statement. A client component is defined by the REMOTE definition statement.

For each TYPE=XTP|XOT LOCAL definition statement, one PCE is allocated.

For each TYPE=XTP REMOTE definition statement that specifies a different IPADDR and/or PORT operand value, one PCE is allocated. Multiple TYPE=XTP REMOTE definition statements can represent the same TCP/IP socket if they share the same IPADDR and PORT operand values but different IFNUM operand values.

For each TYPE=XOT REMOTE definition statement, the VCLMT operand value determines the number of PCEs that are allocated.

One Multi-Channel Link Block (**MCH**) is allocated for each TYPE=XTP|MCH REMOTE definition statement. A TYPE=XTP REMOTE definition statement is used to define an XTP physical X.25 link while a TYPE=MCH REMOTE definition is used to define an XOT logical X.25 link.

For XTP, the MCH represents a physical X.25 interface on the router as identified by the IFNUM operand value. All MCHs on an XTP router share the same TCP/IP socket.

For XOT, the MCH represents a logical X.25 component that can be shared by different TCP/IP sockets (client DTEs). Each XOT virtual circuit on a router uses its own TCP/IP socket.

One MCH Extension Block (**MCHX**) is allocated for each LUNAME entry on a TYPE=XTP|MCH REMOTE definition statement. The MCHX is used to control CTCP PLU to MCH SLU sessions.

One Address Vector Table (**AVT**) is allocated for each TYPE=XTP|MCH REMOTE definition statement. The AVT is used to remember virtual circuit connections on an MCH. The size of the AVT is determined by the VCLMT operand value.

One Virtual Circuit Block (**VCB**) is allocated for each virtual circuit that will access each TYPE=XTP|MCH REMOTE definition statement. The VCB is used to remember virtual circuit state information. The number of VCBs is determined by the VCLMT operand value.

One Logical Unit Block (**LUB**) is allocated for each virtual circuit that will access each TYPE=XTP|MCH REMOTE definition statement. The LUB is used to remember logical unit state information. The number of LUBs is determined by the VCLMT operand value.

The VCLMT operand value of a TYPE=XTP|MCH REMOTE definition statement has a direct bearing on the size of an AVT and the number of VCBs and LUBs that will be created for an MCH. If the VCLMT operand is omitted, a default value is computed using the sum of the entry counts for the PVC, SVC0, SVC3, SVC4 and SVC5 operands.

Note: You should always run a FASTRUN execution of HNAS when ever you make changes to your CDF to determine exactly how much memory will be required for HNAS configuration and run time control blocks.

FTP Server Access Information

This section provides information regarding access instructions for the HNAS FTP Server. You will require an HNAS FTP User name and Password from your support representative. Anonymous logins are not supported.

If you are unable to receive or transfer files via FTP due to your organizations security rules or firewall settings we recommend that *.zip files be sent and received via email attachment.

FTP Server Address

Following is the HNAS ftp server name (the ip address is currently **66.123.107.241** although this address may change should server reassignment occur):

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

FTP Server Login Prompt

The following login prompts are issued by the HNAS FTP Server once a connection occurs (customer ftp user names and passwords are provided by your HNAS Sales or Support representatives):

User (66.123.107.241:(none)):
Password:

FTP Server User Directory Structure

The following ‘user specific’ directories are provided under each FTP user name root directory for customer and HNAS technical support file exchange:

fromsup\\ <- location of edistribution directories or files from support (to user).
\tosup <- location of directories or files to support (from user).

The following HNAS documentation and maintenance subdirectories are listed under all FTP user name root directories:

\hnas_doc <- location of common HNAS documentation files
\hnas*vrm*d <- subdirectories contain VRM specific PDF and web html files
(*vrm*=240 hnas240d)
(see also www.comm-pro.com/hostnas/docs/docindx.htm)

\hnas_maint <- location of common HNAS maintenance files
\hnas*vrm*m <- subdirectories contain VRM specific maintenance information
(*vrm*=240 hnas240m)

(see also www.comm-pro.com/hostnas/maint/index.htm)

Some HNAS **Trial users** are provided with FTP user names and passwords. While these accounts do contain **\tosup** and **\fromsup** subdirectories they do not have access to the **\hnas_doc** or **\hnas_maint** subdirectories. These trial users have the following empty subdirectory that identifies them as trial users:

\@_trial-user_no-doc-or-maint-subdirectory-access

Additional information on FTP Server Links can be found in Chapter 6 (Maintenance) if the HNAS Guide and Reference manual.

FTP Server File Transfer Overview

Once you connect your session it will be located into the root directory (\) associated with your user name. From the root directory you can navigate (Change Directory) through the subdirectories. All directories are READ ONLY except for \tosup which has write access enabling you to transfer files to Comm-Pro, if required. Each FTP user has a unique set of \fromsup and \tosup subdirectories to restrict access to the user name owner.

Prior to downloading files, you will need to issue an FTP change directory command from the root directory to be located into the \fromsup subdirectory. All *.txt files should be transferred in ASCII mode while other file designations (*.bin, *.str and *.zip) must be transferred in binary mode.

Prior to uploading files, you will need to issue an FTP change directory command from the root directory to be located into the \tosup subdirectory. Once you have uploaded the files for Comm-Pro's technical support group please be sure and send an email to your primary HNAS support representative and support@comm-pro.com with a description of the content uploaded.

Please don't hesitate to contact your HNAS support representative should you have any questions or problems accessing the FTP service.

FTP Server Sample Login

Following is a sample line mode login sequence using Windows Command Prompt:

```
C:\>ftp 66.123.107.241
Connected to 66.123.107.241.
220-Serv-U FTP Server v6.4 for WinSock ready...
220-Comm-Pro Associates' FTP Server
220-
220-This server is not intended for public access.
220-There are no programs or data files providing
220-any useful information for the general public.
220-
220-This FTP Server maintains a log of user login,
220-session and file transfer activity. We suggest
220-that you logoff now if you find this policy
220-unacceptable.  The information collected is
220-for security and audit trail purposes only.
220-
220-Please refer to our web site www.comm-pro.com
220-for company, product, support and maintenance
220-information or email support@comm-pro.com with
220-any technical problems or questions that you
220-may have regarding this FTP site or service.
220-
220-cpt-02/14/2002
220 <noticesm.txt>
User (66.123.107.241:(none)): hnas-user-name
331 User name okay, need password.
Password: hnas-password
230 User logged in, proceed.
ftp>
```

The FTP Server is configured to return response **“`331 User name okay, need password.`“** even though the user name is invalid. This is done to provide another layer of security. So please ensure that you are entering the correct user name prior to the password prompt.

FTP Server Sample Download

Following is a sample line mode FTP Server Download:

```
230 User logged in, proceed.
ftp> dir
200 PORT Command successful.
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-   1 user     group           0 Jun 29  2007 .
drw-rw-rw-   1 user     group           0 Jun 29  2007 ..
drw-rw-rw-   1 user     group           0 Oct 17 10:02 fromsup
drw-rw-rw-   1 user     group           0 Aug 21 21:20 hnas_doc
drw-rw-rw-   1 user     group           0 Mar  3  2008 hnas_maint
drw-rw-rw-   1 user     group           0 Oct 17 10:02 tosup
226 Transfer complete.
ftp: 440 bytes received in 0.00Seconds 440000.00Kbytes/sec.
ftp> cd fromsup
250 Directory changed to /fromsup
ftp> dir
200 PORT Command successful.
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-   1 user     group           0 Oct 17 10:02 .
drw-rw-rw-   1 user     group           0 Oct 17 10:02 ..
drw-rw-rw-   1 user     group           0 Oct 17 10:02 history
-rw-rw-rw-   1 user     group     1484230 Jul  9 12:37 hnas_2400082_2008-07-
09_99999_cpt.zip
226 Transfer complete.
ftp: 275 bytes received in 0.00Seconds 275000.00Kbytes/sec.
ftp> binary
200 Type set to I.
ftp> get hnas_2400082_2008-07-09_99999_cpt.zip
200 PORT Command successful.
150 Opening BINARY mode data connection for hnas_2400082_2008-07-
09_99999_cpt.zip (1484230 Bytes).
226 Transfer complete.
ftp: 1484230 bytes received in 2.55Seconds 582.74Kbytes/sec.
```

FTP Server Sample Upload

Following is a sample line mode FTP Server Download:

```
230 User logged in, proceed.
ftp> dir
200 PORT Command successful.
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-   1 user     group           0 Jun 29  2007 .
drw-rw-rw-   1 user     group           0 Jun 29  2007 ..
drw-rw-rw-   1 user     group           0 Oct 17 10:02 fromsup
drw-rw-rw-   1 user     group           0 Aug 21 21:20 hnas_doc
drw-rw-rw-   1 user     group           0 Mar  3  2008 hnas_maint
drw-rw-rw-   1 user     group           0 Oct 17 10:02 tosup
226 Transfer complete.
ftp: 440 bytes received in 0.00Seconds 440000.00Kbytes/sec.
ftp> cd tosup
250 Directory changed to /tosup
ftp> dir
200 PORT Command successful.
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-   1 user     group           0 Oct 17 10:02 .
drw-rw-rw-   1 user     group           0 Oct 17 10:02 ..
drw-rw-rw-   1 user     group           0 Oct 17 10:02 history
226 Transfer complete.
ftp: 181 bytes received in 0.00Seconds 181000.00Kbytes/sec.
ftp> binary
200 Type set to I.
ftp> put cpt_99999_abend_2008-10-01.zip
200 PORT Command successful.
150 Opening BINARY mode data connection for cpt_99999_abend_2008-10-01.zip.
226 Transfer complete.
ftp: 22438672 bytes sent in 18.89Seconds 1187.80Kbytes/sec.
ftp> dir
200 PORT Command successful.
150 Opening ASCII mode data connection for /bin/ls.
drw-rw-rw-   1 user     group           0 Oct 17 11:26 .
drw-rw-rw-   1 user     group           0 Oct 17 11:26 ..
-rw-rw-rw-   1 user     group    22438672 Oct 17 11:26 cpt_99999_abend_2008-
10-01.zip
drw-rw-rw-   1 user     group           0 Oct 17 10:02 history
226 Transfer complete.
ftp: 268 bytes received in 0.00Seconds 268000.00Kbytes/sec.
ftp>
```

CHAPTER 3

Configuration Statement Guide

This page left intentionally blank.

Chapter 3 - Configuration Statement Guide

This chapter provides an overview of the configuration definition statements used by Comm-Pro's X.25 Host Network Access Support (HNAS) software and a description of the steps used to create a session between a remote device and a host PLU.

HNAS configuration definition statements are placed in a Configuration Data File (CDF) that is read by HNAS when it activates. The statements provide:

- the external (static or dynamic) router configurations,
- the types of connections HNAS will accept (PCNE, QLLC, GATE, PAD),
- information telling HNAS how to associate an inbound call with an HNAS SLU,
- information telling HNAS how to set up the SLU to PLU VTAM session,
- information telling HNAS how to direct an outbound call to an XOT/XTP router.

A complete description of HNAS configuration definition statements and their associated operands can be found in Chapter 4 of this document.

HNAS System Overview

HNAS allows a variety of computers to communicate with host systems over X.25 access lines that are connected to Cisco and/or IBM routers. On the flow to the host the routers envelope X25 data in XOT (Cisco) or XTP (IBM) packets which are forwarded to the HNAS host on a TCP/IP network. HNAS removes the XOT or XTP information and forwards the data to a PLU using an application to application VTAM session. On the flow from the host HNAS receives data from the PLU, adds XOT or XTP protocol information and sends the data to the router via the TCP/IP network. The routers strip the XOT/XTP protocol bytes and deliver X25 packets to local router X25 interfaces.

When Cisco's XOT is used there is one TCP session per X25 connection. The association between the TCP session and the X25 interface is established at call setup time via Cisco X25 ROUTE statements. XOT router serial links are logical MCH's.

When IBM's XTP is used there is one TCP session per router. XTP router serial links are physical MCH's, the XTP protocol supports **IFNUM=***serial-interface-number* assignment.

HNAS is an application program containing a number of secondary LUs (SLUs) that perform the same functions as NPSI's virtual LUs. The SLU names are specified by operands in the CDF file. A VTAM Application Major Node File (AMNF) containing the HNAS SLU names must be built and activated before HNAS can communicate with VTAM. For additional information see Application Major Node File, below.

XOT Packetization

On the inbound flow (X.25 -> Router -> HNAS -> PLU), the router encapsulates each X.25 packet (flow control, data, etc.) in an XOT TCP segment. Based on the setting of the MBITCHN= operand of the REMOTE definition statement, segments containing an M-bit chained packet sequence are delivered to the PLU as a series of Only In Chain RUs (one chain per packet) or as an RU chain containing the entire M-bit sequence originally received by the router. The MBITCHN= operand (TYPE=MCH REMOTE) controls the M-bit chain to RU chain mapping process The RU size is determined from the PLU's BIND.

On the outbound flow (PLU -> HNAS -> Router -> X.25), HNAS builds XOT TCP segments containing full X.25 packets linked by M-bits. An M-bit chain represents an RU chain element or an RU chain depending on the setting of the MBITCHN= operand of the REMOTE definition statement.

The X.25 packet size is determined at session startup from facilities contained in the Call Request packet. The XOT protocol is described in RFC1613.

Information on **Cisco router** definitions for HNAS environments can be found in **Appendix C** of the HNAS VnRnMn Guide and Reference manual.

XTP Packetization

On the inbound flow (X.25 -> Router -> HNAS -> PLU), the router collects an M-bit packet chain and forwards it to HNAS in a single XTP TCP segment. HNAS delivers the segment to the PLU as an RU chain or as a series of Only In Chain RUs based on the setting of the MBITCHN operand of the REMOTE definition statement. No record of X.25 M-bits or data packet boundaries is carried in the segment created by the XTP router.

On the outbound flow (PLU -> HNAS -> Router -> X.25), HNAS builds an XTP TCP segment for each RU chain element or for each RU chain based on the setting of the MBITCHN= operand of the REMOTE definition statement. The router builds an M-bit packet sequence for each XTP TCP segment received from HNAS.

NPSI Emulation Support

HNAS supports most NPSI functions including PCNE (LLC0), QLLC (LLC3), GATE (LLC4) and PAD (LLC5). The LLC value is determined from call user data byte 0 (CUD0) using standard NPSI values. Configuration parameters allow overriding NPSI values as well as selecting the LLC via a subaddress digit (the last digit in the called address in a call request packet). The CUD0 value is also commonly referred to as the PID (protocol Identifier).

The following table associates the CUD0 or PID (protocol identifier) value with the standard NPSI LLC component:

CUD0 to LLC Type Table (Standard NPSI Values)

CUD0 Values	LLC Type	NPSI Component
X’01’,X’41’,X’51’,X’81’	LLC5	PAD
X'C0', X'CC'	LLC0	PCNE
X'C2', X'CA'	LLC2	PSH (not supported)
X'C3'	LLC3	QLLC
X'C4'	LLC4	GATE
X'CB'	LLC3	Boundary Network Node (BNN) QLLC
X'E3'	LLC3	Subarea Network Node (SNN) QLLC (not supported)
X'EB'	LLC3	Intermediate Network Node (INN) QLLC (not supported)

NPSI MCH to HNAS Logical MCH Migration (XOT Environment)

When migrating from NPSI to HNAS in an XOT environment there are some important differences to take into consideration. NPSI has definitions include statements defining physical X25 Multi-Channel Links (MCHs). With HNAS calls received on an X25 link attached to a serial interface on a Cisco router are encapsulated with XOT protocol bytes and routed to a TCP/IP interface by ROUTE statements in the router. Each XOT session is carried in a single TCP/IP session. HNAS has no knowledge of the physical MCHs attached to the router. Thus, if you need to test or later use more than one X.25 line on your router, it probably will be necessary to recreate in HNAS what we call **logical MCHs** which correspond one-for-one with NPSI **physical MCHs**. This is typically done by using the ***called*** or ***calling*** DTE address field from incoming Call Request packets and inserting the ***calling*** DTE address in outgoing Call Request packets as follows:

For incoming Call Request packets, HNAS can use the regular ***called*** DTE address if it is presented to the router by the network. If it is not, as is the case in France with Transpac, HNAS can use a mapped DTE address (configured in the router’s routing table) that is inserted in front of the ***called*** DTE address field (i.e., before any subaddress, if provided). HNAS can also route calls to logical MCHs using the ***calling*** DTE address. For additional information see RTEIN= operand description (TYPE=XOT REMOTE statement).

In outgoing Call Request packets, HNAS can insert the ***calling*** DTE address (from the DCEADDR= operand) for non-GATE calls and for GATE calls if this address is not inserted by the CTCP (the GATE application). This ***calling*** DTE address is used in the routing table of the router to route the call out to the appropriate serial interface. The ***calling*** DTE address, if it has been inserted by HNAS and the network does not like it, has to be removed either by a configuration command on the interface (x25 suppress-calling-address) or by the routing table itself.

Either ***calling*** or ***called*** addresses can be used for address filtering or substitution although there's usually more flexibility (digits available) in the HNAS callout originating calling address.

Starting with 220, a variety of options are available to associate or filter *calling* or *called* DTE addresses using SVCn=, LLCn= or RTEOUT= parameters. The one exception is that only a single DTEADDR= either on the REMOTE TYPE=MCH or TYPE=MXT can be used amongst all three (dteaddr1-dteaddr2-dteaddr3) values coded for Callout Connection Routing support.

See Chapter 4 of the HNASBook in the REMOTE definition statement, the DCEADDR= and OPTIONS= operand values: REPDCEADDR and STRIPRTEIN for more explanation.

NPSI to HNAS LU Name Migration

We recommend using different LU names for HNAS than are being used with NPSI.

For normal GATE (CONNECT=NO or omitted on the NPSI X.25.MCH macro or on the HNAS TYPE=MCH REMOTE definition statement), the MCH control session LUs are defined by the LUNAME= operand in NPSI as well as in HNAS. For NPSI, the data session LUs are defined in a switched major node as a PU/LU pair. For HNAS, the data session LUs are identified in the SVC4= operand of the TYPE=MCH REMOTE definition statement.

For GATE Fast Connect (CONNECT=YES|SUBD|CUD0 on the NPSI X.25.MCH macro or on the HNAS TYPE=MCH REMOTE definition statement), the MCH control session LUs are defined by the LUNAME= operand in NPSI as well as in HNAS. The LU names used for data sessions are defined in the NPSI configuration by the PRFLU= and SUFFIX= on the X25.VC statement. For HNAS, the data session LU names are defined using suboperands of the LUNAME= operand of the TYPE=MCH REMOTE definition statement.

Note: All HNAS LUs are application LUs and must be defined by APPL statements in an Application Major Node File (AMNF). See Chapter 2 for information on generating the AMNF using the PARM=FASTRUN operand.

If you need to use the same LU names for HNAS and NPSI and NPSI is currently using these LUs then you will need to do the following:

For normal GATE, inactivate the MCH line and VARY release the associated PU and inactivate the associated Switched Major Nodes.

For GATE Fast Connect, inactivate the MCH line and release the PU, inactivate the Fast Connect lines of NPSI and VARY release the associated PUs.

NPSI SWNODE Migration to HNAS AMNF Considerations

For NPSI, VTAM parameters are normally specified on LU statements in the NPSI Switched Major Node File (SMNF). For HNAS, these same VTAM parameters may be specified on the APPL statements in the HNAS AMNF see “AMNF Considerations”, above).

If you will be running the HNAS FASTRUN process to create the HNAS AMNF, the VTAM parameters can be coded on the REMOTE definition statements in the HNAS CDF that identify LU resources. The FASTRUN process will then propagate the parameters to the APPL statements in the AMNF. The FASTRUN process does not check the VTAM parameters for validity of value or context. All non-HNAS parameters are treated as VTAM parameters and are propagated from the CDF to the AMNF 'asis'. This means that incorrectly coded HNAS parameters may be placed in the AMNF.

If you will be creating the HNAS AMNF manually (not recommended), the VTAM parameters for the NPSI LUs must be specified on the respective APPL statements in the HNAS AMNF.

Some NPSI SMNF parameters like DISCNT=, which can appear on a PU statement, are not supported in the HNAS AMNF because HNAS has no physical services component. If DISCNT= is specified on an APPL statement, VTAM will generate an error message.

The MAXOUT= NPSI SMNF parameter (PU statement) is used to control processing of the first message received from an LLC0 or LLC5 device. MAXOUT=6 prevents NPSI from sending the message to the SSCP. Other MAXOUT= values cause NPSI to sent the first message to the SSCP as a logon message. HNAS LUs do not have an SSCP session. The first message will be processed as a logon message (using USS tables or Interpret tables) if the MCHSOL HNAS component is selected when the APPLNAME= parameter is processed. For additional information see “XOT PCNE (LLC0) and PAD (LLC5) Callin Resource Definitions” in this chapter and the description of the REMOTE statement’s APPLNAME= parameter in Chapter 4.

HOST Application Terminal Definitions

Some host application program (e.g. CICS, IMS) have their own terminal definitions used to specify session characteristics (bracket usage, send and receive RU sizes, etc.). If definitions are created for HNAS SLUs be sure to use parameters that match those used for NPSI sessions.

Application Major Node File (AMNF)

HNAS SLU names are specified by operands in the CDF. Each name must appear in an active VTAM AMNF in order for HNAS to operate. If directed to do so, see FASTRUN in Chapter 2, HNAS can produce the AMNF. Comm-Pro strongly recommends this method of generating the AMNF. An AMNF has the following form:

```
HNASAPPL VBUILD TYPE=APPL
SLU0001  APPL  DLOGMOD=INTERACT,EAS=1
.....
SLU000n  APPL  DLOGMOD=INTERACT,EAS=1
```

In the above, INTERACT is a default DLOGMOD used if HNAS produces the AMNF. EAS=1 specifies that only one session can be active with each HNAS SLU. Default DLOGMOD values can be overridden by coding DLOGMOD=name on a TYPE=MCH or TYPE=XTP REMOTE statement. This will replace INTERACT in all APPL statements generated for PCNE, PAD and GATE data session LUs created for the REMOTE. If different LLC types require different DLOGMOD values or if a different DLOGMOD is required for the GATE control session LU then the NAS generated AMNF must be edited.

Definition Statement (CDF) Overview

The following pages show the basic configuration definition statements required to configure HNAS systems supporting the various LLC types. Only the most significant operands are shown. Before attempting to configure a HNAS system, be sure to review all the operands associated with each definition statement (see Chapter 4). The BUILD and END definition statements, required in all HNAS configurations, are not shown in the examples.

XOT Definition Summary

Cisco routers using the XOT protocol create a TCP session for each X25 call. In addition to defining the router configuration, definition statements route calls to a logical MCHs (Multi Channel Links) where NPSI functions are defined.

The following summarizes the most significant XOT configuration statements.

```
         BUILD ...
```
Provides system wide definitions and options (number of virtual circuits, buffer sizes, etc.).

```
SRVR     LOCAL TYPE=XOT IPADDR=ip-addr PORT=1998
               RTEIN=(mch-name/dte-addr,...)
               RTEOUT=(rtr-name/dte-addr,...)
```
Provides the HNAS IP address, routing information used to direct inbound calls to logical MCHs and outbound calls to a specific IP address. Routing can be based on the called or calling DTE addresses which can contain wildcard characters. Outbound calls may be balanced across a group of IP addresses (BALANCERTEOUT OPTION) and additional paths can be tried if an outbound call fails. Inbound calls can be distributed to a collection of MCHs if the BALANCERTEIN OPTION is specified. Multiple LOCAL statements are used if multiple IP addresses are to be supported.

```
RNCNIN    REMOTE TYPE=XOT IPADDR=DYNAMIC PORT=DYNAMIC VCLMT=limit
```
Creates ‘limit’ call in resource control blocks for use by any XOT router. Use this statement if you wish to accept XOT calls from any XOT router.

```
R1CNIN    REMOTE TYPE=XOT IPADDR=ip-addr PORT=DYNAMIC VCLMT=limit
```
Identifies an XOT IP address and creates ‘limit’ call in resource control blocks for the address. Use this statement to dedicate a pool of call in control blocks to a single router with the given IP address.

```
R1CNOT    REMOTE TYPE=XOT IPADDR=ip-addr PORT=1998 VCLMT=limit
                 HOME=local-name
```
Identifies an XOT IP address and creates ‘limit’ call out or call in resource control blocks for exclusive use of router with the given IP address. This type of remote statement is required if HNAS is to call out (send XOT call request packets). HOME is required if there is more than one LOCAL statement.

```
MCH1      REMOTE TYPE=MCH SVC0=(...) SVC3=(...) SVC4=(...) SVC5=(...)
                 APPLNAME=(plu-nm1,...,plu-nmn) HOME=local-name
```
Defines a logical MCH and it’s associated resources. The SVCi= parameters specify the number and type of switched virtual circuits on the MCH and provide the SLU names that the PLU will communicate with. APPLNAME= provides the names of PLUs that HNAS can request sessions with. Other parameters are similar to those used by NPSI: GATE=, MBITCHN=, CTCP=, CUD0=, LOGTAB=, USSTAB=, etc. HOME= is required if there is more than one LOCAL statement and the MCH is used for call out.

```
SLUMXT    REMOTE TYPE=MXT CUD=... FAC=... DCEADDR=...
```
Defines an MCH extension which can be addressed by SVC0/5= SLU definition strings. The

extension allows a callout SLU to have unique call user data, facilities and *calling* DTE address.

Inbound XOT calls are routed to an MCH by the LOCAL TYPE=XOT RTEIN= parameter in conjunction with the called or calling DTE address in the X25 call request packet.

Outbound calls (callout SLU bound by a PLU or a GATE CTCP delivers a call request pack to HNAS) are routed to an IP address by the LOCAL TYPE=XOT RTEOUT= parameter in conjunction with the called or *calling* DTE address in the call request packet.

```
PU01      REMOTE TYPE=SPU,
                 IDBLK=xxx,IDNUM=xxx
                 LUNAME=(,nane1,name2,...)
```

Defines a remote PU and LU configuration that will use the QLLC protocol to communicate with HNAS. Components in HNAS activate the remote's components (ACTPU, ACTLU, etc.). The PLU communicates with the named LUs.

```
          END
```

Final HNAS definition statement

The following examples provide an overview. Be sure to consult Chapter 4 for definitions of all operands associated with each definition statement. Build and END statements are not shown.

XOT PCNE (LLC0) and PAD (LLC5) Callin Resource Definitions

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4       ; HNAS IP ADDRESS
               PORT=1998                ; WELL KNOWN XOT PORT
               RTEIN=(MCH1/3106,        ; 3106 & 4106 GO TO MCH1  <230>
                      MCH1/4106,MCH2)   ; ALL OTHERS GO TO MCH2.  <230>
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE 16 PORTS FOR 192.40.60.3 TO INITIATE SESSIONS ON.
*        CODE IPADDR=DYNAMIC TO ALLOW ANY ROUTER TO CALL IN.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNIN   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.3       ; ROUTER IP ADDRESS
               PORT=DYNAMIC             ; ROUTER INITS CONNECT
               VCLMT=16                 ; NUMBER OF PORTS

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE 2 LOGICAL MCH LINKS FOR CISCO XOT ROUTER ACCESS
*        TO PCNE (LLC0) AND PAD (LLC5) SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH               ; IDENTIFY LOGICAL XOT MCH
               DLOGMOD=USER001         ; FOR APPL AMNF STATEMENTS GEN'ED
                                       ; FOR MCH1 BY FASTRUN
               SVC0=(3,                ; 2 LLC0 SLUS FOR SVCS
                     M1LU001/X123450-X123460I1,;CALL WITH CUD1-3=123450
                                       ; OR 123460 GETS SLU M1LU001
                                       ; AND PLU NAMED PLU1.
                     M1LU002,          ; ALL OTHERS GET SLU M1LU002 OR 3,
                     M1LU003)          ; PLU SET BY USS TABLE CMD LOGIC
                                       ; (LOGON APPLID(...))
               SVC5=(2,M1LU501,M1LU502); 2 LLC5 SLUS FOR SVCS
               PAD=INTEG               ; HNAS PROVIDES LLC5 PAD SUPPORT
               TRAN=EVEN               ; LLC5 ASCII TRANSLATE TABLES
               USSTAB=ISTINCDT         ; IBM USS TABLE
               APPLNAME=(MCHSOL,       ; 0=HNAS USS CMD TABLE PROCESSOR
                         PLU1)         ; 1=PLU1

MCH2     REMOTE TYPE=MCH               ; IDENTIFY XOT MCH
               SVC0=2                  ; 2 LLC0 SVCS.  HNAS ASSIGNS SLU
                                       ; NAMES OF MCH20001 & MCH20002
               SVC5=2                  ; 2 LLC5 SVCS.  HNAS ASSIGNS SLU
                                       ; NAMES OF MCH25001 & MCH25002
               PAD=INTEG               ; HNAS PROVIDES LLC5 PAD SUPPORT
               TRAN=EVEN               ; LLC5 ASCII TRANSLATE TABLES
                                       ; SLUS GET PLU VIA SYSL=. SUBADDR
                                       ; GENERATES APPLNAME= INDEX.
               SYSL=(SUBD=1/0,SUBD=09/1) ; SUBADDR=1 -> 0 -> PLU1,
                                       ;  SUBADDR=09 -> 1 -> HNAS CONSOLE
               APPLNAME=(PLU1,CONSOLE) ; 0=PLU1 , 1=HNAS CONSOLE ROUTINE
```

The XOTSRVR TYPE=XOT LOCAL definition statement provides the HNAS IP address and routing information for inbound calls. As coded, inbound call requests with a *called* DTE address starting with 3106 or 4106 will be routed to MCH1. All other inbound calls will be routed to MCH2 (because no DTE address was coded). If a call cannot be routed to a logical MCH it is cleared.

The R1CNIN REMOTE statement creates the control blocks required to manage 16 XOT sessions initiated by the router with IP address 192.40.60.3. Code IPADDR=DYNAMIC to allow calls from any XOT router.

When an XOT call request is received the following steps are taken:

If the *called* DTE address starts with 3106 or 4106 then MCH1 provides definitions and resources for the call. All other calls are routed to MCH2.

For MCH1:

If CUD0 is X'C0' or X'CC' then an LLC0 session is requested. If CUD0 = X'01', X'41', X'51', or X'81' then an LLC5 session is requested. Any other CUD0 value causes HNAS to clear the call.

After the LLC type is established, an SLU control block for the session is located using information from the SVC0= or SVC5= parameters (based on the selected LLC). Control block allocation is a 2 step process. First the SVC0= or SVC5= definitions are searched for an available SLU configured with a hex digit string or DTE address matching the call request packet's IDNUM value in call user data bytes or the *calling* DTE address. If a match is found the SLU control block is allocated (removed from an available queue, placed on an active queue). If there is no match, the SVC0= or SVC5= definitions are searched for an available SLU configured without a hex digit string or DTE address. If one is found it is allocated. If not the call is cleared. In the above, SLU M1LU001 can only be used by a caller with CUD1-3=X'123450' or X'123460'. M1LU002 and M1LU003 can be allocated to any inbound LLC0 call. M1LU501 and M1LU502 can be allocated to any inbound LLC5 call.

After SLU allocation HNAS determines the PLU name that will be used for the session. There are 3 sources for the PLU name:

1) APPLNAME= index coded with the SLU name. When the second component in the SLU's definition in the SVC0= or SVC5= parameter ends with Id (I=Inbound, d=decimal digits) then the SLU is reserved for inbound calls and the PLU is selected by using the digit(s) following the I as a select index in APPLNAME=. In the above M1LU001 has 'I1' following the hex digit string (remote's IDNUM). The '1' (following the 'I') used as an index in APPLNAME=, produces the PLU name of PLU1 (this PLU is always used by M1LU001).

2) If the SLU definition does not include an APPLNAME= index then the SYSL= parameter is used to produce an APPLNAME index from subaddress digit(s) or a call user data byte. If SYSL= is omitted or fails to produce an index then the first APPLNAME= entry is selected.

3) If 1) or 2) selects MCHSOL (reserved word in APPLNAME=) then the PLU name is selected by data from the remote in conjunction with an interpret table (LOGTAB=, not used above) or a USS Table (USSTAB=ISTINCDT, is the default IBM table).

In the above, M1LU002 or MLLU003 do not include an APPLNAME= index. MCH1 does not have a SYSL= parameter so the first APPLNAME= operand is used as the default PLU name. In this example the default PLU is MCHSOL. MCHSOL directs HNAS to produce the PLU name using data from the remote processed against an interpret table (LOGTAB=, not used in this example) or against a USS Table (USSTAB=ISTINCDT). USS Messages, if defined, are sent. With the ISTINCDT table the remote is expected to enter LOGON APPLID(xxx) to identify the PLU.

After the SLU and PLU names are known, HNAS sends a REQSESS to VTAM to request a session with the PLU. If the REQSESS triggers a BIND from the PLU, then PLU to SLU to remote device communication can proceed. If a BIND is not received the call is cleared.

When MCHSOL is used, HNAS sends a call accept so that the LOGON message can be entered. If the PLU name comes from the APPLNAME= parameter then the call accept is sent when HNAS receives the PLU's BIND.

For MCH2:

Processing for MCH2 is similar to that for MCH1. When a call is directed to MCH2 the LLC type is determined from standard CUD0 values.

No SLU names are coded on the SVC0= or SVC5= parameters. When this is done the names are generated from the first 4-characters of the REMOTE name (padded on the right with '#' characters if necessary) followed by the character '0' and a 3-digit hexadecimal SVC0= operand index number or the character '5' followed by the 3 digit hexadecimal SVC5= index number. These SLUs are for inbound traffic only. An APPLNAME= PLU select index cannot be specified when HNAS generates the SLU names. The generated names for MCH2 are MCH20001, MCH20002 (SVC0=2) and MCH25001, MCH25002 (SVC5=2).

Since the PLU name is not known after SLU allocation, the SYSL= operand is used to generate an APPLNAME= index from the call request packet's subaddress digits (last 1 or 2 digits in the *called* DTE address) or a call user data byte. For callers routed to MCH2, subaddress 1 selects PLU1 and subaddress 09 selects the HNAS console command processor. CONSOLE, like MCHSOL, is a reserved name in the APPLNAME= parameter. When CONSOLE is the 'PLU' the remote is in session with the HNAS console services routine. This allows the remote to communicate directly with HNAS. The available commands are described the 'HNAS Console Subsystem and Operations Guide'.

If PLU1 is selected HNAS sends a REQSESS to VTAM to start the session.

Since MCHSOL does not appear in MCH2's APPLNAME= parameter, interpret and USS tables are not be used.

XOT PCNE and PAD Callout Resource Definitions

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4       ; HNAS IP ADDRESS
               PORT=1998                ; XOT WELL KNOWN PORT
               RTEOUT=(R1CNOT/3106,R2CNOT/3106);CALLS TO 3106 USE R1/2CNOT
                       R3CNOT)          ; ALL ADDRS MATCH THIS ENTRY
                                        ; (NO DTE ADDR CODED)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT CALLOUT PORTS (3 IP ADDRESSES, 4 PORTS EACH)
*        THESE PORTS CAN ALSO BE USED FOR CALLIN.
*        NO INBOUND CALLS WILL BE ACCEPTED (NO RTEIN= ON LOCAL).
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNOT   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.10      ; ROUTER IP ADDRESS
               PORT=1998                ; HNAS CAN INITIATE CONNECT.
               VCLMT=4                  ; CONNECTIONS ALLOWED WITH IP ADDR

R2CNOT   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.11      ; ROUTER IP ADDRESS
               PORT=1998                ; HNAS CAN INITIATE CONNECT
               VCLMT=4                  ; CONNECTIONS ALLOWED WITH IP ADDR

R3CNOT   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.12      ; ROUTER IP ADDRESS
               PORT=1998                ; HNAS CAN INITIATE CONNECT
               VCLMT=4                  ; CONNECTIONS ALLOWED WITH IP ADDR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH LINK FOR CISCO XOT ROUTER ACCESS
*        TO PCNE (LLC0) AND PAD (LLC5) CALLOUT RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH                ; IDENTIFY XOT MCH
               SVC0=(2,                 ; 2 LLC0 SLUS FOR SVCS
                                        ; NEXT SLU HAS 2 CALLED ADDRS
                     M1LU0001/31060001-31070001O,
                     M1LU0002/47280001O);
               SVC5=(2,                 ; 2 LLC5 SLUS FOR SVCS
                     M1LU5001/31067770O//11223344, NO MXT, CUD FOR SLU
                     M1LU5002/67459999O/MXT1);     MXT FOR SLU

               PAD=INTEG                ; HNAS PROVIDES LLC5 PAD SUPPORT
               TRAN=EVEN                ; LLC5 ASCII TRANSLATE TABLES
               HOME=XOTSRVR             ; NOT REQUIRED (1 LOCAL STATEMENT)

MXT1     REMOTE TYPE=MXT,CUD=44000000,DCEADDR=9999 ; CALL USER DATA AND
                                        ; CALLING DTE ADDRESS FOR M1LU5002
```

The XOTSRVR TYPE=XOT LOCAL definition statement provides the HNAS IP address and routing information used to direct outbound calls to specific IP addresses. As coded, outbound call requests with a *called* DTE address starting with 3106 will be routed to the R1CNOT or the R2CNOT REMOTE. All other outbound calls will be routed to the R3CNOT REMOTE because no DTE address was coded with R3CNOT. If ‘S’ is coded following a DTE address in a RTEOUT= entry then the source (calling) address is used for address compare against the entry. For LLC0 and LLC5 calls the called address comes from the SVC0= or SVC5= TYPE=MCH REMOTE statement operand. The calling address comes from the DCEADDR= operand (not used in this example).

The R1CNOT, R2CNOT and R3CNOT REMOTE statements create the control blocks required to allow HNAS to initiate sessions at three IP addresses (up to 4 per address).

MCH1 defines 2 callout SVC0 resources and 2 callout SVC5 resources. The distinguishing feature of a callout resource definition is the character ‘O’ following the DTE address in the SVC0= or SVC5= parameter. This flags the SLU as a callout SLU. Up to 3 DTE addresses (separated by ‘-’s) may be specified following the SLU name.

MXT1 defines call user data and a *calling* DTE address for LLC0 and LLC5 SLUs that address the MXT1 MXT. In this example only M1LU5002 refers to the MXT1 REMOTE. When the M1LU5002 LU is bound the HNAS call request packet will contain the call user data and *calling* DTE address specified by the MXT1 REMOTE. Up to 63 bytes of call user data may be specified with CUD= on a TYPE=MCH or TYPE=MXT REMOTE. The definition string for M1LU5001 specifies no MXT (‘//’) and call user data (‘11223344’). When call user data is specified this way only 4 bytes may be coded. M1LU0001 and M1LU0002 have no associated call user data so the HNAS default (X’01000000’) will be sent. The FAC= operand is not coded so all LLC0/5 call requests generated by HNAS will contain default facilities of X’0101420909430404’ (see FAC= operand description in Chapter 4).

When HNAS builds a call request packet the called and calling addresses, call user data and facilities fields must be known. For NPSI, these values are obtained by parsing the DIALNO= parameter on a PATH statement in a switched major node. The called and calling addresses are part of DIALNO=. Indexes in the DIALNO= string can reference an X25.OUFT statement which supplies facilities and call user data (OPTFACL=, USRFILD= and USRFIL2=). HNAS obtains the called address from the SVC0/5= operand. The calling address, facilities and call user data are provided by DCEADDR=,FAC= and CUD=. By referencing TYPE=MXT REMOTE statements in the SVC0/5= string it is possible for each callout LU to have a unique calling address, facilities and call user data.

Callout operation starts when a PLU acquires the HNAS callout SLU (HNAS receives a BIND). HNAS builds a call request packet and locates a REMOTE (i.e. IP address) to send the packet to. Leading digits in the called or calling address are compared to the digits in elements of the RTEOUT= list. If a match occurs the named REMOTE is addressed, an available port is located and the call request is sent to the remote’s IP address. A RTEOUT= entry with no DTE digits matches all addresses (R3CNOT, above).

If a call accept is returned to an HNAS call request, then communication between the remote and the PLU may begin.

Starting with 230, if the call is not accepted (clear received or timeout occurs) then HNAS continues looking in the RTEOUT= string for address matches. This allows a DTE address to be tried at different routers. If the end of the RTEOUT is reached with out a successful call then secondary DTE addresses are tried in the RTEOUT= list. If all addresses fail, the VTAM ACB in the HNAS SLU is closed (this generates a NOTIFY for the PLU).

With the above configuration, when MLU0001 is bound the RTEOUT= parameter is scanned in order to find an IP address to use for the first *called* DTE address (31060001). The R1CNOT REMOTE is selected because the first RTEOUT= entry (R1CNOT/3106) specifies R1CNOT if the called (target) address starts with 3106. If R1CNOT is idle (INIT=IDLE coded), if all 4 ports are in use or if the HNAS call fails then processing of the RTEOUT= list continues and port allocation will be tried on R2CNOT and then R3CNOT. If the end of the RTEOUT= parameter is reached with no call being accepted then the second *called* DTE address for M1LU0001 (31070001) is selected and processed through the RTEOUT= list entries. The maximum number of called addresses that may be coded on an LLC0 or LLC5 SLU is 3.

Starting with 220, if OPTIONS=(BALANCERTEOUT) is coded on the XOTSRVR LOCAL statement, consecutive entries in the RTEOUT list with the same DTE address are considered to be a round robin group (in the above, R1CNOT and R2CNOT). All members of the group are searched for an available port. However, each time the group is used the starting point for the search is moved. In the above, the first call will use R1CNOT, the second call will use R2CNOT and the third call will use R1CNOT. This tends to balance the use of ports on R1CNOT and R2CNOT. Without this option ports on R2CNOT are used only if no ports are available on R1CNOT.

XOT GATE (LLC4) Callin and Callout Resources Definitions

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4      ; HNAS IP ADDRESS
               PORT=1998               ; XOT WELL KNOWN PORT
               RTEIN=(MCH1/3106,MCH2)  ; CALLED ADDR 3106 TO MCH1 ALL
                                       ; OTHERS TO MCH2 (NO DTE ADDR,
                                       ; MCH2 NOT SHOWN)
               RTEOUT=(R1CNOT/3106T    ; SELECT BASED ON CALLED DTE ADDR
                       R2CNOT/3107S    ; SELECT BASED ON CALLING DTE ADDR
                       R3CNOT)         ; NO DTE -- OK FOR ALL CALLS


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT CALLIN PORTS FOR USE BY 192.40.60.10
*        CODE IPADDR=DYNAMIC TO ALLOW ANY ROUTER TO CALL IN.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNIN   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.10      ; ROUTER IP ADDRESS
               PORT=DYNAMIC             ; ROUTER INITIATES CONNECT
               VCLMT=4                  ; CONNECTIONS ALLOWED AT IP ADDR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT CALLOUT PORTS (3 IP ADDRESSES, 4 PORTS EACH)
*        THESE PORTS WILL BE USED FOR CALLIN IF ALL R1CNIN PORTS
*        ARE USED.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNOT   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.10      ; ROUTER IP ADDRESS
               PORT=1998                ; HNAS CAN INITIATE CONNECT
               VCLMT=4                  ; CONNECTIONS ALLOWED AT IP ADDR
R2CNOT   REMOTE TYPE=XOT IPADDR=192.40.60.11 PORT=1998 VCLMT=4
R3CNOT   REMOTE TYPE=XOT IPADDR=192.40.60.12 PORT=1998 VCLMT=4


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH LINK FOR CISCO XOT ROUTER ACCESS
*        TO GATE (LLC4) CALLIN RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH                  ; IDENTIFY XOT MCH
                GATE=GENERAL              ; MCH SUPPORTS GATE
                SUBADDR=YES               ; SUBADDRESS CAN SET LLC
                LLC4=(4)                  ; SUBADDRESS=4 SETS LLC4
*                                           ANY OTHER SUBADDRESS REQUIRES
*                                           CUD0/CTCP MAPPING
                SVC4=(2,M1LU401,M1LU402)  ; 2 LLC4 DATA SESSION SLUS
*                                           INBOUND & OUTBOUND CALLS
                LUNAME=(CTCP1/PLU1,CTCP2) ; SLU NAMES FOR GATE CONTROL
*                                           SESSIONS
                CUD0=(00,01,C0,C1)        ; CUD0 TO CTCP MAPPING
```

```
CTCP=(00,00,01,01)          ; OVERRIDE CUD0=ALL VALUES
HOME=XOTSRVR                ; NOT REQUIRED (1 LOCAL STMT)
```

The TYPE=XOT LOCAL definition statement provides the HNAS IP address and information for routing inbound calls (by *called* DTE address) to a logical HNAS MCH and outbound calls (by called or *calling* DTE address) to a specific IP address.

As coded, inbound call requests with a *called* DTE address starting with 3106 will be routed to MCH1. All others inbound calls will be routed to MCH2 (not shown).

Outbound calls (routed by RTEOUT=) will use ports defined by R1CNOT if the called (target) DTE address starts with 3106 and ports defined by R2CNOT if the calling (source) DTE address starts with 3107. R3CNOT (with no DTE address) is used for all other calls. The round robin group logic (see PCNE and LLC5 Callout Resource Definitions for XOT, above) may also be used. Care must be taken when coding RTEOUT= because the DTE addresses that will be used by the CTCP must be known. If a call fails (timeout or clear response to HNAS call request) then a clear is sent to the CTCP. If attempts with multiple DTE addresses are desired, the CTCP must provide them.

The R1CNOT, R2CNOT and R3CNOT REMOTE statements provide control blocks for callout sessions with the specified IP addresses.

GATE Definition Parameters

GATE=GENERAL allows definition of LLC4 resources. It also allows parameters like SUB-ADDRESS=YES and LLC*i*= which permit LLC selection by subaddress digit (*i*=0|3|4|5). Additionally, GATE=GENERAL permits the CUD0= and CTCP= operands which allow over-riding of the standard CUD0 to LLC type mappings. CUD0 LLC selection only occurs if sub-address LLC selection fails to select an LLC type (LLC4 in this example).

SVC4= defines the GATE SLU data session LU names. These SLUs may be used for inbound and outbound GATE calls.

LUNAME= defines the GATE SLU control session LU names. The control session is used to establish data sessions. When the control session SLU name is followed by a PLU name (CTCP1/PLU1) HNAS issues a REQSESS to VTAM to request a session with the PLU. This activates the control session as soon as HNAS starts. If the PLU name is not coded, HNAS waits for a BIND to activate the control session. Some CTCPs make only one attempt to acquire the control session SLU. If the CTCP is brought up before HNAS this attempt will fail. Coding the PLU name in HNAS ensures that the control session is started when the HNAS job is started after the CTCP job.

CUD0= and CTCP= revise the standard values used to establish the LLC type and to select a CTCP based on CUD0 (see GATE Callin Processing, below).

The control and data session SLU names must be defined APPL statements in a VTAM application major node.

Note: When LLC4 is established via subaddress selection (SUBADDR=YES,LLC4=(*list*)), the VC is always connected to the first CTCP from the LUNAME= operand. The CUD0= and CTCP= operands are not used.

Note: When subaddress selection (SUBADDR=YES,LLC*i*=(*list*)) is in effect but no match results in any LLC*i*=(*list*) operand, LLC type and CTCP selection are performed using the CUD0 byte in conjunction with the CUD0= and CTCP= operands.

GATE Callin Processing

When a call is received for MCH1, the subaddress (last digit of *called* DTE address) is first examined because SUBADDR=YES is specified. If the subaddress digit is 4, LLC4 is set and the first CTCP in the LUNAME= operand is used for the VC connection (SLU=CTCP1, PLU=PLU1). If the subaddress is any value other than 4 (or if the *called* DTE address is not present), the LLC is determined from call user data byte 0 (CUD0). For GATE the default LLC4 values, which may be obtained by coding CUD0=ALL, are X'02', X'C4' (first CTCP) and X'00', X'03-X'2F', NULL (second CTCP). These values are overridden by the above CUD0= and CTCP= operands. For MCH1 CUD0=X'00' and X'01' set LLC4 and select the first CTCP (CTCP1). CUD0=X'C0' and X'C1' set LLC4 and select the second CTCP (CTCP2). The mapping between CUD0= values and CTCP= values (which are indexes in the LUNAME= operand) is by position. As with NPSI, specification of 80, 83 or 85 in the CTCP operand sets the LLC type to 0, 3 or 5.

After LLC and CTCP selection HNAS locates an available LLC4 SLU created by the SVC4= operand and sends the call request to the CTCP using the selected control session. If the CTCP responds with a call accept on the control session, HNAS forwards it to the remote and initiates the data session by sending a REQSESS to VTAM. When the CTCP BINDs the data session GATE callin session establishment is complete. If no LLC4 SLU is available or if a BIND is not received, the call is cleared.

Some CTCPs expect the Resource ID carried in an incoming call request packet to identify a specific SLU. HNAS creates resource ids based on the position of the SLU name in the SVC4= operand offset by the number of PVCs defined. If there are no PVCs and If there are 3 SVC4= SLUs, the resource ids will be 1 through 3. If there are 2 PVCs defined, the resource ids will be 3 through 5. These considerations are important if the CTCP has a table mapping resource ids to SLU names.

GATE Callout Processing

When the CTCP sends a call request to HNAS using the control session, a data session LU is allocated and the call is routed based on the called or *calling* DTE addresses in the CTCP's call request packet and the RTEOUT= operand (for more on routing for this MCH see the RTEOUT= description, above). If a call accept is returned it is sent to the CTCP on the control session. The data session is started by sending a REQSESS to VTAM requesting a session with the CTCP. A bind from the CTCP to the HNAS SLU completes session establishment.

XOT GATE Fast Connect Resources Definitions

In order to make the above GATE configuration a fast connect (FC) configuration the following operands would be added or changed in the MCH1 REMOTE TYPE=MCH definition:

CONNECT=CUD0

Specifies that the MCH is a GATE fast connect MCH. All MCH resources must be LLC4. When CUD0 is specified there are multiple CTCPs selected by CUD0 values in connection with the CUD0= and CTCP= operands (these are required in this case).

```
LUNAME=(CTCP1/CPLU1/AAA/3/2,    ; SLU NAMES: AAA00003, AAA00004
        CTCP2//BBB//2)          ; SLU NAMES: BBB00000, BBB00001
```

For GATE fast connect, LUNAME= specifies the CTCP SLU names (CTCP1 and CTCP2, above) and optionally the associated PLU names (CPLU1 for CTCP1) **and** the data session SLU names that will be in permanent session with each CTCP.

CTCP1 will request a session with CPLU1. Data session SLU names start with AAA, the starting sequence number is 3 and 2 SLUs are defined.

CTCP2 waits for a bind (no PLU name specified). Data session SLU names start with BBB, the first sequence number is 0 (default value) and 2 SLUs are defined.

All SLU names must appear in APPL statements in a VTAM application major node.

Call processing for GATE fast connect differs from non-fast connect in that the data sessions are permanently active. Call request and clear packets (to or from the CTCP) use the data session LUs (not the control session LUs as with non-FC gate).

When an inbound call is received an SLU is located by searching for a bound SLU that currently has no X25 session (i.e. is not attached to a VC).

Outbound calls are routed to an IP address using the RTEOUT= parameter and the called or *calling* DTE address from the call request packet.

XOT QLLC (LLC3) Callin Resource Definitions

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4      ; HNAS IP ADDRESS
               PORT=1998               ; HNAS PORT NUMBER
               RTEIN=(MCH1/3106,       ; DNIC=3106 ROUTED TO MCH1
                      MCH2)            ; ALL OTHERS TO MCH2 (NOT SHOWN)


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CALLIN RESOURCES.
*        8 PORTS FOR 192.40.60.3 TO CALL IN ON.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNIN   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.3      ; ROUTER IP ADDRESS
               PORT=DYNAMIC            ; ROUTER INITS CONNECT
               VCLMT=8                 ; CONNECTIONS ALLOWED

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE LOGICAL MCH LINK FOR QLLC SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH         ; IDENTIFY XOT MCH
               USSTAB=ISTINCDT   ; STANDATRD IBM USS TABLE
               SVC3=(2,          ; REMOTE TYPE=SPU CANDIDATES TO
                     PU01,PU02)  ; CHECK THESE FOR IDBLK/IDNUM IN
                                 ; A NEW CALL

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CONFIGURATIONS MATCHED TO INBOUND QLLC CALLS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PU01    REMOTE TYPE=SPU            ; QLLC REMOTE PU DESCRIPTION
               IDBLK=201 IDNUM=00001 ; PU SELECTED BY
                                   ; IDBLK/IDNUM IN XID FROM RMT
               APPLNAME=MCHSOL     ; GET PLU NAME FROM REMOTE LU
               LUNAME=(,           ; NO SLU FOR LOCADDR=1
                       LU0101)     ; SLU FOR LOCADDR=2

PU02    REMOTE TYPE=SPU            ; QLLC REMOTE PU DESCRIPTION
              IDBLK=201 IDNUM=00002 ; PU SELECTED BY IDBLK/IDNUM
                                   ; IN XID FROM REMOTE PU
              APPLNAME=(PLU1,CICS) ; PLU NAMES
               SYSL=(DATA=PLU1/0,  ; INPUT DATA PLU1 OR XXX CREATES
                    DATA=XXX/1)    ; INDEX IN APPLNAME=
              LUNAME=(,LU0201,LU0202) ; SLUS FOR LOCADDR=2 AND 3
```

The TYPE=XOT LOCAL statement provides the HNAS XOT IP address and routing information for inbound calls. Inbound calls with a *called* DTE address starting with 3106 are routed to MCH1. All others are routed to MCH2 (not shown).

The R1CNIN REMOTE statement creates the control blocks required to manage 8 XOT sessions initiated by 192.40.60.3.

MCH1 REMOTE TYPE=MCH defines a logical MCH. When a call is routed to MCH1 the following occurs:

The LLC TYPE is determined from CUD0. Since only LLC3 resources are defined (by SVC3=) X'C3', X'CB', X'E3' and X'EB' are the only valid CUD0 values for MCH1. These values select QLLC processing.

XOT QLLC PU/LU Activation

After the LLC type is set to LLC3 HNAS brings up the remote PU / SSCP session (HNAS plays the role of the SSCP and activates the PU). An XID is received from the remote which provides the PU type and the IDBLK/IDNUM values identifying the remote.

The REMOTE TYPE=SPU statements addressed by MCH1's SVC3= parameter are searched in order to find an available REMOTE TYPE=SPU resource with a matching IDBLK/IDNUM. If no SPU is located the call is cleared. The SVC3= parameter provides a candidate list to be used when a QLLC call is received. The name of a TYPE=SPU remote may appear in any number of REMOTE TYPE=MCH SVC3= operands.

Starting with 230, all SPUs in the CDF are searched for an IDBLK/IDNUM match even if they are not referenced in the SVC3= operand of the target MCH.

When the TYPE=SPU REMOTE has been located HNAS knows the configuration on the calling PU. The LUs in the LUNAME= list are activated (ACTLU sent). Once an LU is active PLU selection is made when data is received on the LU/SSCP (HNAS) session. The data is used in conjunction with SYSL=(DATA=...) operands to generate an index in the APPLNAME= parameter which selects a PLU name.

For PU01 (SYSL= omitted) the received message is passed to the HNAS USS Table processor which finds the PLU name using an interpret table (LOGTAB= operand, not used in this example) or a USS Table. USSTAB=ISTINCDT specifies the default IBM table which expects a LOGON APPLID(...) command from the remote.

For PU02 characters starting with the first non-blank character in the received message are compared with the DATA= operands in the SYSL= operand. A match generates an index in the APPLNAME= parameter which, in turn, provides the PLU name. As coded, 'PLU1' generates an index of zero which selects PLU1 in the APPLNAME= parameter. 'XXX' selects CICS as the PLU name (via index=1). If no match is found in the SYSL=(DATA=...) operands the input data is passed the HNAS interpret table / USS table processor.

When the PLU name is known, a REQSESS is sent by HNAS to VTAM. The BIND response is forwarded to the remote to start the QLLC session.

XOT QLLC Callout Resource Definitions (CLOTINITYP= initiated callout)

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4      ; HNAS IP ADDRESS
               PORT=1998               ; XOT WELL KNOWN PORT
               RTEOUT=(R1CNOT/3106,R2CNOT/3106);CALLS TO 3106 USE R1/2CNOT
                       R3CNOT)         ; ALL ADDRS MATCH THIS ENTRY
                                       ; (NO DTE ADDR CODED)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT CALLOUT PORTS (3 IP ADDRESSES, 4 PORTS EACH)
*        THESE PORTS CAN ALSO BE USED FOR CALLIN.
*        NO INBOUND CALLS WILL BE ACCEPTED (NO RTEIN= ON LOCAL).
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.10     ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS CAN INITIATE CONNECT.
               VCLMT=4                 ; CONNECTIONS ALLOWED WITH IP ADDR

R2CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.11     ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS CAN INITIATE CONNECT
               VCLMT=4                 ; CONNECTIONS ALLOWED WITH IP ADDR

R3CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.12     ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS CAN INITIATE CONNECT
               VCLMT=4                 ; CONNECTIONS ALLOWED WITH IP ADDR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH LINK FOR CISCO XOT ROUTER ACCESS
*        TO QLLC (LLC3) CALLOUT RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH               ; IDENTIFY XOT MCH
                SVC3=(2)               ; ALLOW 2 LLC3 FOR SVCS
                USSTAB=ISTINCDT        ; STANDATRD IBM USS TABLE
                HOME=XOTSRVR           ; NOT REQUIRED (1 LOCAL STATEMENT)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CONFIGURATIONS MATCHED TO QLLC CALLS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PU01    REMOTE TYPE=SPU            ; QLLC REMOTE PU DESCRIPTION
               IDBLK=201 IDNUM=00001 ; PU SELECTED BY
*                                  ; IDBLK/IDNUM IN XID FROM RMT
               APPLNAME=MCHSOL     ; GET PLU NAME FROM REMOTE LU
               LUNAME=(,           ; NO SLU FOR LOCADDR=1
                       LU0101)     ; SLU FOR LOCADDR=2
               OPTIONS=CLOTINITYP=BIND  ; INITIATE CALL WHEN LU0101 BOUND
               CUD=C3000000        ; CALLOUT USER DATA
               DCEADDR=9999        ; CALLING DTE ADDRESS
```

```
                 DTEADDR=31067770     ; CALLED DTE ADDRESS
                 FAC=0101420808430404 ; CALLOUT FACILITIES

PU02    REMOTE TYPE=SPU              ; QLLC REMOTE PU DESCRIPTION
                 IDBLK=201 IDNUM=00002 ; PU SELECTED BY IDBLK/IDNUM
                                      ; IN XID FROM REMOTE PU
                 APPLNAME=(PLU1,CICS) ; PLU NAMES
                 SYSL=(DATA=PLU1/0,    ; INPUT DATA PLU1 OR XXX CREATES
                       DATA=XXX/1)     ; INDEX IN APPLNAME=
                 LUNAME=(,LU0201,LU0202) ; SLUS FOR LOCADDR=2 AND 3
                 OPTIONS=(CLOTINITYP=TIMER, ; INITIATE CALL WHEN MCHTMR= FOR
*                                          ; MCH1 EXPIRES
                          CLOTFAILRTYLMT=4, ; LIMIT CALL FAILURES TO 3
                          CLOTCONLMT=5) ; LIMIT TIMER CONNECTIONS TO 5
                 CUD=C3000000         ; CALLOUT USER DATA
                 DCEADDR=9999         ; CALLING DTE ADDRESS
                 DTEADDR=67459999     ; CALLED DTE ADDRESS
                 FAC=0101420808430404
```

The XOTSRVR TYPE=XOT LOCAL definition statement provides the HNAS IP address and routing information used to direct outbound calls to specific IP addresses. As coded, outbound call requests with a *called* DTE address starting with 3106 will be routed to the R1CNOT or the R2CNOT REMOTE. All other outbound calls will be routed to the R3CNOT REMOTE because no DTE address was coded with R3CNOT. If ‘S’ is coded following a DTE address in a RTEOUT= entry then the source (calling) address rather than the target (called) address is used for address compare against the entry.

When OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE is coded on a TYPE=SPU REMOTE definition statement, the *called* DTE address, *calling* DTE address, facilities and call user data for an outbound LLC3 call comes from the DTEADDR=, DCEADDR=, FAC= and CUD= operands, respectively, on the same TYPE=SPU REMOTE definition statement. Up to 63 bytes of facilities and call user data may be specified with FAC= and CUD= operands. Note that you must specify facilities for QLLC callout SPUs because the Cisco router requires them and a default value is not provided (see FAC= operand description in Chapter 4).

When HNAS builds a Call Request packet, the called and calling addresses, facilities and call user data fields must be known. For NPSI, these values are obtained by parsing the DIALNO= parameter on a PATH statement in a switched major node. The called and calling addresses are part of DIALNO=. Indexes in the DIALNO= string can reference an X25.OUFT statement which supplies facilities and call user data (OPTFACL=, USRFILD= and USRFIL2=). HNAS obtains these values from the DTEADDR=, DCEADDR=, FAC= and CUD= operand, respectively.

The R1CNOT, R2CNOT and R3CNOT REMOTE statements create the control blocks required to allow HNAS to initiate sessions at three IP addresses (up to 4 per address).

MCH1 is configured to allow 2 inbound LLC3 connections (SVC3=2). If a callout SPU is not already connected, it can be allocated to an inbound SVC that is directed to MCH1 (by RTEIN=) based on IDBLK/IDNUM matching.

An outbound Call Request for the SPU named **PU01** is attempted when any SLU in its LUNAME= operand (LU0101 is the only SLU defined) is bound (acquired). Since the CLOTFAILRTYLMT= option is not specified, a default value of 3 is used. Since the CLOTCONLMT= option is also not specified, a default value of 0 is used (unlimited Call Requests allowed). BIND initiated callout processing continues until a Call Accept is received or the CLOTFAILRTYLMT= value is reached.

For the first condition (Call Accept received), additional callout attempts will be performed after the current call is cleared anytime a BIND is received. We recommend that for CLOTINITYP=BIND, you always omit the CLOTCONLMT= option so that no restrictions are placed on the number of BIND initiated Call Requests.

For the second condition (CLOTFAILRTYLMT=3 value reached), the BIND that initiated the Call Request is ended in error and the ACB for the SLU is closed. Note that if multiple SLUs were defined in the LUNAME= operand and one or more had active BINDs outstanding, all would be ended in error and all ACBs would be closed. Subsequent BIND initiated call requests are blocked. This is an indication that something is wrong with the *called* DTE address, *calling* DTE address, facilities and/or call user data. The MRMT command can be used to correct the offending value(s) followed by the VARY ACT command to manually retry the call.

An outbound Call Request for the SPU named **PU02** is attempted when the OPTIONS=MCHTMR= value expires for the *first* MCH in the CDF. In this configuration, MCH1 is the first and only MCH and a default MCHTMR= value of 60 seconds is used. This timeout supplies the event that starts the callout process. Timer initiated callout processing continues until a Call Accept is received or the CLOTFAILRTYLMT= value is reached.

For the first condition (Call Accept received), 5 successful Call Requests are allowed because CLOTCOMLMT=5 was specified. Additional Call Requests are blocked. In this case, the VARY ACT command must be used to initiate another the call. Note that the VARY ACT command resets the CLOTCONLMT= count so another 4 Call Requests will be started automatically by timeout. If you wish to restrict new Call Requests further, use MRMT to change the CLOTCONLMT= value or set CLOTINITYP=CONSOLE. For the latter, VARY ACT will then be the only way to initiate a callout.

For the second condition (CLOTFAILRTYLMT=4 value reached), further Call Requests are blocked. This is an indication that something is wrong with the *called* DTE address, *calling* DTE address, facilities and/or call user data. The MRMT command can be used to correct the offending value(s) followed by the VARY ACT command to manually retry the call.

When a callout operation starts, HNAS builds a Call Request packet and locates a TCP/IP REMOTE (i.e., IP address) to send the packet to. Leading digits in the called or calling address are compared to the digits in elements of the RTEOUT= list. If a match occurs the named REMOTE is addressed, an available port is located and the call request is sent to the remote’s IP address. A RTEOUT= entry with no DTE digits matches all addresses (R3CNOT, above).

If a Call Accept is returned to an HNAS Call Request, communication between the remote and the SPU may begin. Refer to the section above titled '**XOT QLLC PU/LU Activation** on page 3-20.

Starting with 230, if the call is not accepted (clear received or timeout occurs), HNAS continues looking in the RTEOUT= string for address matches. This allows a DTE address to be tried at different routers. If the end of the RTEOUT= is reached without a successful call, the call is treated as a failure and is subject to CLOTFAILRTYLMT= counting.

Using the sample configuration above, a callout is initiated for PU01 when LU0101 is bound. The RTEOUT= parameter is scanned in order to find an IP address to use for the *called* DTE address associated with PU01 (31067770). The R1CNOT REMOTE is selected because the first RTEOUT= entry (R1CNOT/3106) specifies R1CNOT if the called (target) address starts with 3106. If R1CNOT is idle (INIT=IDLE coded), if all 4 ports are in use or if the HNAS call fails then processing of the RTEOUT= list continues and port allocation will be tried on R2CNOT and then R3CNOT. A callout is initiated for PU02 when the MCHTMR= expires on MCH1. Since the called (target) DTE address is 67459999, only R3CNOT can be used for the callout attempt. R1CNOT and R2CNOT are associated with DNIC 3106 which eliminates there use for DNIC 6745. R2CNOT has no DTE association so may be used for any DTE addresses.

General Notes for the CLOTINITYP= option.

1) **CLOTINITYP=BIND** should be specified when the host application is configured to acquire the SLUs (OPNDST OPTCD=ACQ). This will cause HNAS to establish the connection to the SPU when any of the SLUs in the LUNAME= operand are bound.

2) **CLOTINITYP=TIMER** should be specified when the SLUs on the SPU are able to logon to one or more applications. In this case MCHSOL must be coded as an application name within the APPLNAME= operand and a valid USS table name must be specified in the USSTAB= operand. In this case HNAS will establish the connection to the SPU when the OPTIONS=MCHTMR=*value* expires for the *first* defined MCH.

3) **CLOTINITYP=CONSOLE** can be specified if you wish to establish the connection to the SPU manually using the HNAS console subsystem. In this case you will have to issue the VARY command to the SPU as follows: **VARY *spuname* ACT**

XOT QLLC Callout Resource Definitions (SVC3= timer initiated callout)

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4      ; HNAS IP ADDRESS
               PORT=1998               ; XOT WELL KNOWN PORT
               RTEOUT=(R1CNOT/3106,R2CNOT/3106);CALLS TO 3106 USE R1/2CNOT
                       R3CNOT)         ; ALL ADDRS MATCH THIS ENTRY
                                       ; (NO DTE ADDR CODED)

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT CALLOUT PORTS (3 IP ADDRESSES, 4 PORTS EACH)
*        THESE PORTS CAN ALSO BE USED FOR CALLIN.
*        NO INBOUND CALLS WILL BE ACCEPTED (NO RTEIN= ON LOCAL).
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.10     ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS CAN INITIATE CONNECT.
               VCLMT=4                 ; CONNECTIONS ALLOWED WITH IP ADDR

R2CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.11     ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS CAN INITIATE CONNECT
               VCLMT=4                 ; CONNECTIONS ALLOWED WITH IP ADDR

R3CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.12     ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS CAN INITIATE CONNECT
               VCLMT=4                 ; CONNECTIONS ALLOWED WITH IP ADDR

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH LINK FOR CISCO XOT ROUTER ACCESS
*        TO QLLC (LLC3) CALLOUT RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH               ; IDENTIFY XOT MCH
                SVC3=(2,               ; 2 LLC3 SPUS FOR SVCS
                     PU01/31067770O,   ; NO MXT                  <230>
                     PU02/67459999O/MXT1); MXT FOR SPU           <230>
                FAC=0101420808430404   ; CALLOUT FACILITIES
                USSTAB=ISTINCDT        ; STANDATRD IBM USS TABLE
                HOME=XOTSRVR           ; NOT REQUIRED (1 LOCAL STATEMENT)

MXT1     REMOTE TYPE=MXT
                CUD=C3000000C8D5C1E2   ; CALL USER DATA
                DCEADDR=9999           ; CALLING DTE ADDRESS FOR PU02

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CONFIGURATIONS MATCHED TO QLLC CALLS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PU01    REMOTE TYPE=SPU             ; QLLC REMOTE PU DESCRIPTION
               IDBLK=201 IDNUM=00001 ; PU SELECTED BY
```

```
                              ; IDBLK/IDNUM IN XID FROM RMT
              APPLNAME=MCHSOL ; GET PLU NAME FROM REMOTE LU
              LUNAME=(,       ; NO SLU FOR LOCADDR=1
                     LU0101)  ; SLU FOR LOCADDR=2

PU02    REMOTE TYPE=SPU            ; QLLC REMOTE PU DESCRIPTION
        IDBLK=201 IDNUM=00002      ; PU SELECTED BY IDBLK/IDNUM
                                   ; IN XID FROM REMOTE PU
        APPLNAME=(PLU1,CICS)       ; PLU NAMES
         SYSL=(DATA=PLU1/0,        ; INPUT DATA PLU1 OR XXX CREATES
              DATA=XXX/1)          ; INDEX IN APPLNAME=
        LUNAME=(,LU0201,LU0202)    ; SLUS FOR LOCADDR=2 AND 3
```

The XOTSRVR TYPE=XOT LOCAL definition statement provides the HNAS IP address and routing information used to direct outbound calls to specific IP addresses. As coded, outbound call requests with a *called* DTE address starting with 3106 will be routed to the R1CNOT or the R2CNOT REMOTE. All other outbound calls will be routed to the R3CNOT REMOTE because no DTE address was coded with R3CNOT. If ‘S’ is coded following a DTE address in a RTEOUT= entry then the source (calling) address rather than the target (called) address is used for address compare against the entry. For LLC3 calls the called address comes from the SVC3= operand on the TYPE=MCH REMOTE statement. The calling address comes from the DCEADDR= operand (supplied by the MXT in this example).

The R1CNOT, R2CNOT and R3CNOT REMOTE statements create the control blocks required to allow HNAS to initiate sessions at three IP addresses (up to 4 per address).

MCH1 defines 2 callout SVC3 resources. The distinguishing feature of a callout resource definition is the character ‘O’ following the DTE address in the SVC3= parameter. This flags the SPU as a callout SPU.

Call Requests for callout SPUs are attempted automatically when the OPTIONS=MCHTMR= value expires for an MCH. For MCH1 above, a default MCHTMR= value of 60 seconds is used. Automatic calling continues indefinitely at the MCHTMR= interval until a Call Accept is received.

MXT1 defines call user data and a *calling* DTE address for the LLC3 SPUs that address the MXT1 MXT. In this example only PU02 refers to the MXT1 REMOTE. When the PU02 SPU Call Request is created, it will contain the call user data and *calling* DTE address specified by the MXT1 REMOTE. Up to 63 bytes of call user data may be specified with CUD= on a TYPE=MCH or TYPE=MXT REMOTE. The definition string for PU01 specifies no MXT (‘//’). PU01 has no associated call user data so the HNAS default (X’C3000000’) will be sent. The FAC= operand is coded on the MCH1 MCH so all LLC3 Call Requests generated by HNAS for this MCH will contain the specified facilities of X’0101420808430404’. Note that you must specify facilities for QLLC callout SPUs either on the MCH or associated MXT because the Cisco router requires them and a default value is not provided (see FAC= operand description in Chapter 4).

When HNAS builds a Call Request packet the called and calling addresses, call user data and facilities fields must be known. For NPSI, these values are obtained by parsing the DIALNO= parameter on a PATH statement in a switched major node. The called and calling addresses are part of DIALNO=. Indexes in the DIALNO= string can reference an X25.OUFT statement which supplies facilities and call user data (OPTFACL=, USRFILD= and USRFIL2=). HNAS obtains the called address from the SVC3= operand. The calling address, facilities and call user data are provided by DCEADDR=,FAC= and CUD=. By referencing TYPE=MXT REMOTE statements in the SVC3= string it is possible for each callout SPU to have a unique calling address, facilities and call user data.

Callout operation starts when the MCHTMR= value expires for the root MCH. HNAS builds a Call Request packet and locates a TCP/IP REMOTE (i.e., IP address) to send the packet to. Leading digits in the called or calling address are compared to the digits in elements of the RTEOUT= list. If a match occurs the named REMOTE is addressed, an available port is located and the call request is sent to the remote's IP address. A RTEOUT= entry with no DTE digits matches all addresses (R3CNOT, above).

If a Call Accept is returned to an HNAS call request, communication between the remote and the SPU may begin. Refer to the section above titled '**XOT QLLC PU/LU Activation** on page 3-20.

Starting with 230, if the call is not accepted (clear received or timeout occurs), HNAS continues looking in the RTEOUT= string for address matches. This allows a DTE address to be tried at different routers. If the end of the RTEOUT= is reached with out a successful call, the call is retried during the next MCHTMR= interval.

Using the sample configuration above, when the MCHTMR= expires, the RTEOUT= parameter is scanned in order to find an IP address to use for the *called* DTE address associated with PU01 (31067770). The R1CNOT REMOTE is selected because the first RTEOUT= entry (R1CNOT/3106) specifies R1CNOT if the called (target) address starts with 3106. If R1CNOT is idle (INIT=IDLE coded), if all 4 ports are in use or if the HNAS call fails then processing of the RTEOUT= list continues and port allocation will be tried on R2CNOT and then R3CNOT. For PU02, since the called (target) DTE address is 67459999, only R3CNOT can be used for the callout attempt. R1CNOT and R2CNOT are associated with DNIC 3106 which eliminates there use for DNIC 6745. R2CNOT has no DTE association so may be used for any DTE addresses.

Multiple HNAS Server Resource Definitions

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVERS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR1 LOCAL TYPE=XOT IPADDR=192.40.60.4; FIRST HNAS IP ADDRESS
               RTEIN=MCH1               ; INBOUND CALLS USE MCH1
               RTEOUT=R1CNOT            ; OUTBOUND CALLS USE R1CNOT

XOTSRVR2 LOCAL TYPE=XOT IPADDR=192.40.60.5; SECOND HNAS IP ADDRESS
               RTEIN=MCH2               ; INBOUND CALLS USE MCH2
               RTEOUT=R2CNOT            ; OUTBOUND CALLS USE R2CNOT

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT CALL IN PORTS.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RTRCNIN  REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.99      ; PORTS FOR A SPECIFIC ROUTER
               PORT=DYNAMIC             ; CALLS MAY ARRIVE AT EITHER LOCAL
               VCLMT=4                  ; NUMBER OF CALL IN RESOURCES

GBLCNIN  REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTERS
               IPADDR=DYNAMIC           ; ALL ROUTERS MAY CONNECT
               PORT=DYNAMIC             ; CALLS MAY ARRIVE AT EITHER LOCAL
               VCLMT=16                 ; NUMBER OF CALL IN RESOURCES

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CALL OUT PORTS AND ROUTER IP ADDRESSES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNOT   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.11      ; ROUTER (DESTINATION) IP ADDRESS
               PORT=1998                ; HNAS INITIATES CONNECT
               VCLMT=4                  ; OUTBOUND CONNECTIONS ALLOWED
               HOME=XOTSRVR1            ; LOCAL STMT (SOURCE IP ADDRESS)

R2CNOT   REMOTE TYPE=XOT                ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.12      ; ROUTER (DESTINATION) IP ADDRESS
               PORT=1998                ; HNAS INITIATES CONNECT
               VCLMT=4                  ; OUTBOUND CONNECTIONS ALLOWED
               HOME=XOTSRVR2            ; LOCAL STMT (SOURCE IP ADDRESS)


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH LINKS FOR CISCO XOT ROUTER ACCESS.
*        EACH MCH HAS A CALL IN AND CALLOUT LLC0 RESOURCE.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH SVC0=(2,M1LU0001/31060001O,M1LU0002)
               HOME=XOTSRVR1            ; USE XOTSRVR1 RTEOUT=

MCH2     REMOTE TYPE=MCH SVC0=(2,M2LU0001/31060001O,M2LU0002)
               HOME=XOTSRVR2            ; USE XOTSRVR2 RTEOUT=
```

The XOTSRVR1 and XOTSRVR2 LOCAL statements define 2 IP addresses that XOT routers may use to communicate with HNAS. The RTEIN= operands direct inbound calls to unique logical MCHs. This is not a requirement, LOCAL statements may use the same MCH or MCHs for inbound calls. The RTEOUT= operands identify TYPE=XOT REMOTE statements used to provide the socket control block and the IP address of the XOT router that receives a HNAS call request. Call out ports are connected to a particular LOCAL statement by the HOME= operand (see R1CNOT and R2CNOT REMOTE statements). When performing call out operations (BIND delivered to LLC0/LLC5 call out LU (e.g. M1LU0001) or call request received from a GATE CTCP) HNAS must know the source and destination IP addresses for the TCP/IP packet that carries the XOT call request packet. The source address is provided by the LOCAL TYPE=XOT statement addressed by the HOME= operand on the TYPE=MCH REMOTE statement. The destination address is provided by a TYPE=XOT REMOTE statement addressed by the RETOUT= operand on the TYPE=XOT LOCAL statement. For XOT, HOME= operands are only required when there is more than one TYPE=XOT LOCAL statement.

The RTRCNIN REMOTE statement creates a call in socket pool for use by the router with IP address 192.40.60.99. Calls from this router may arrive with either HNAS LOCAL IP address.

The GBLCNIN REMOTE statement creates a call in socket pool for use by any XOT router. Calls from routers may arrive with either HNAS LOCAL IP address. The search for a call in port is conducted in definition statement order. The IPADDR=DYNAMIC statement should follow the call in pools for specific addresses so that calls from specific addresses to not consume resources in the DYNAMIC pool.

The R1CNOT and R2CNOT TYPE=XOT REMOTE statements with PORT=1998 create a pool of call out ports used when HNAS initiates a session. These statements provide the destination IP address for HNAS calls. The source IP address for the call comes from the LOCAL statement which is addressed by the HOME= operand on the remote statement. The ports defined by these statements may be used for call in sessions arriving at the IP address of the LOCAL the REMOTE is attached to (by the HOME= operand).

PVC Resource Definitions

Example HNAS CDF XOT PVC Definition

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4      ; HNAS IP ADDRESS
               PORT=1998               ; WELL KNOWN XOT PORT NUMBER
               RTEIN=(...)             ;
               RTEOUT=(...)            ;

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE 4 CALLIN AND 4 CALLOUT PORTS FOR 192.40.60.3
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNIN   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.3      ; ROUTER IP ADDRESS
               PORT=DYNAMIC            ; ROUTER INITS CONNECT
               VCLMT=4                 ; CONNECTIONS ALLOWED

R1CNOT   REMOTE TYPE=XOT               ; IDENTIFY XOT ROUTER
               IPADDR=192.40.60.3      ; ROUTER IP ADDRESS
               PORT=1998               ; HNAS INITS CONNECT
               OPTIONS=(PVCSETPTMR=20,PVCRECONTMR=30)
                                       ; SETUP & VTAM CONNECT TIMERS
               VCLMT=4                 ; CONNECTIONS ALLOWED

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE LOGICAL MCH LINK FOR CISCO XOT ROUTER ACCESS
*        SUPPORTING LLC0, LLC3 AND LLC5 PVC SESSIONS
*        (HNAS DEFAULT PACKET SIZE 256 (0808) AND WINDOW SIZE 4 (0404)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH                    ; IDENTIFY XOT MCH
                PVC=(3,                     ; DEFINE 2 PVCS
                    PVC001/0/0/3/SERIAL0-2, ; SLU-NM, LLC, APPL-IDX,
                                            ; LCN, REMOTE INTERFACE NAME
                    PVC002/5/1/4/SERIAL0-2/R1CNOT/PVCMXTWP)
                    SPU001/3/2/255/SERIAL0-2/R1CNOT/PVCMXTWP)
                                            ; SLU-NM, LLC, APPL-IDX,
                                            ; LCN, REMOTE INTERFACE NAME,
                                            ; XOT REMOTE FOR HNAS SEND
                                            ; OF PVC SETUP AND ASSOCIATED
                                            ; MXT-NAME DEPICTING THE HNAS
                                            ; PACKET-SIZE AND WINDOW-SIZE.
               PAD=INTEG                    ; INTEGRATED PAD SUPPORT
               TRAN=EVEN                    ; EVEN PARITY ASCII TRANSLATE
               APPLNAME=(PLU1,PLU2)         ; PLU NAMES

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE MXT FOR OPTIONAL WINDOW AND PACKET SIZE VALUES
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PVCMXTWP REMOTE TYPE=MXT                    ; IDENTIFY XOT MXT
```

```
                FAC=(420707,                ; SET PACKET SIZE TO 128
                     430202)                ; SET WINDOW SIZE TO 2

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CONFIGURATIONS MATCHED TO PVC QLLC SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
SPU001  REMOTE TYPE=SPU              ; QLLC REMOTE PU DESCRIPTION
               IDBLK=201             ; PU SESSION VALIDATED BY
               IDNUM=00001           ; IDBLK/IDNUM IN XID FROM RMT
               APPLNAME=MCHSOL       ; GET PLU NAME FROM REMOTE LU
               LUNAME=(,             ; NO SLU FOR LOCADDR=1
                       SLU00102)     ; SLU FOR LOCADDR=2
...
```

Example Cisco Router XOT PVC Configuration

```
!
interface Serial0/2
x25 win 2                   ! the x25 win|wout|ips|ops values are
x25 wout 2                  ! default interface values used for PVCs
x25 ips 128                 ! and SVCs and will not appear in the
x25 ops 128                 ! show run display.  The x25 facility
...                         ! windowsize|packetsize values are not
...                         ! used for PVCs.
...
x25 ltc 4                   ! first three lcns are for PVC use.
x25 pvc 1 xot 192.40.60.3 interface Serial MCH1 pvc 1
    packetsize 256 256 windowsize 4 4    ! override vc defaults
x25 pvc 2 xot 192.40.60.3 interface Serial MCH1 pvc 2
x25 pvc 3 xot 192.40.60.3 interface Serial MCH1 pvc 3
...
   In this Cisco example for interface Serial 0/2 pvc 1-3 are
   associated with HNAS MCH1.
```

The XOT protocol uses PVC SETUP packets to establish PVC sessions. The SETUP is sent by an initiator and responded to by a responder. The packets contain, among other things, the initiator's name and logical channel number and the responder's name and logical channel number. The SETUP will fail if the router and HNAS do not have the same configuration information for the PVC. Cisco routers use interface names like SERIAL0, SERIAL0/2 and SERIAL0/1/2. Since '/' is used as an HNAS operand separator, the HNAS convention is to use '-' where a '/' is required in an interface name. The HNAS interface name consists of the characters 'Serial' followed by the MCH name on the TYPE=MCH REMOTE. The HNAS interface name is required when the PVC configuration is defined to the router.

In the above, the LOCAL TYPE=XOT statement provides the HNAS IP address and routing information. No HNAS routing (RTEIN= or RTEOUT=) is required for PVCs -- the target MCH is carried in an inbound PVC SETUP packet and, if HNAS is to send a PVC SETUP, the name of the TYPE=XOT remote to use is in the PVC definition string (R1CNOT in the

PVC002 and SPU001 definitions). The frequency of SETUP attempts is controlled by the PVCSETUPTMR OPTIONS parameter.

The R1CNIN and R1CNOT REMOTE statements define ports that HNAS uses for TCP/IP sessions carrying PVC traffic. An inbound port is required for each PVC session initiated by the remote and an outbound port is required for each PVC session initiated by HNAS.

HNAS PVC definition summary:

Each PVC definition string in the HNAS **PVC=** operand contains the following definition fields separated by '/'s:

1) the HNAS SLU name for LLC0 or LLC5 PVCs or the HNAS SPU name for LLC3 PVCs (note that SLU names must be in an application major node APPL statement)
2) the LLC type for the session
3) an index into the APPLNAME= string so that HNAS knows the PLU name for the session (note that for LLC3 (QLLC) PVCs, the index is not used so 255 is specified as a place holder)
4) the HNAS logical channel number (arbitrary, but the router must be configured with it)
5) the remote's serial interface name (for PVC001 it's SERIAL0/1, coded as SERIAL0-1)
6) optionally, an HNAS router name. If this name is provided HNAS will send a PVC SETUP packet using the named TYPE=XOT remote. If the name is omitted, HNAS waits for a PVC SETUP from the remote.
7) optionally, an HNAS MXT name. If this name is provided HNAS will send a PVC SETUP packet with the provided window and packet size values taken from the FAC= operand facility 43 and 42, respectively. Note that HNAS currently defaults to a window size of 4 and a packet size of 256.

Cisco router PVC definition summary:

Each X25 PVC (XOT) definition string under the Cisco router Interface Serial*n* or Serial*n/n* **X25 PVC...** definition for HNAS XOT contains the following fields:

x25 pvc *interfaceserialpvc#* **xot** *hnas-ipaddr* **interface serial** *mch-name* **pvc** *mchpvc#* *option*

1) *interfaceserialpvc#* is the PVC number of the connecting device on the serial interface.
2) *hnas-ipaddr* field represents the IP address of the XOT host that you are connecting to.
3) *mch-name* field represents the destination HNAS MCH *mch-name*.
4) *mchpvc#* field represents the destination pvc number under the respective MCH. HNAS PVC numbers are assigned relative to 1 based upon the order coded under each MCH.
5) *option* fields allow for specific connection features such as packetsize and windowsize.

Please refer to the Cisco Wide-Area Networking Command Reference manual for additional information concerning x25 pvc (XOT) syntax and coding rules.

After the PVC SETUP and PVC SETUP response have been exchanged, For LLC0 or LLC5 PVCs, HNAS sends a REQSESS to VTAM to request a BIND from the PLU identified by the third parameter in the PVC definition string (APPLNAME= index). When the BIND is received data exchange may begin. If data is received before the BIND is received a RESET is sent to the remote.

For LLC3 PVCs, HNAS sends an XID request to the SPU in order to validate the IDBLK/IDNUM values associated with the SPU. The values specified for the named TYPE=SPU REMOTE must match the values for the real remote SPU, otherwise the PVC connection will not operate properly. A Reset with CAUSE/DIAG=000/080, DIAGX=0B will also be generated.

Note: Starting with 230 (APAR 2300107), the PVC windows size and packet size values can now be coded via an MXT. Please refer to the Remote TYPE=MCH PVC= operand in Chapter 4 for additional information.

PVC Operational Notes

Once the Cisco router and HNAS PVCs are defined properly, the XOT PVC set-up sequence will complete without errors regardless of the state of the physical X.25 serial interface. The router sends a PVC Setup followed by a 0F reset (network operational) whether or not there is an active link. When HNAS sends a PVC Reset to the router there will be a Reset Confirm whether or not there is an active link.

The HNAS LU ACB associated with a PVC is opened when the setup exchange between the router and HNAS completes successfully. After the ACB has been opened HNAS will issue a REQSESS to ask the PLU for a BIND (if the PLU name is specified in PVC= definition string) or will wait for the PLU to acquire the HNAS SLU (PLU name not specified). If the REQSESS sent by HNAS fails (NAS3702W alert) then, after APAR 2400053, the REQSESS is retried once per minute (prior to the APAR longer retry times were used). After APAR 2400059 the frequency of VTAM session connect attempts is controlled by the OPTIONS=PVCRECONTMR=secs parameter.

If HNAS is configured to send SETUP packets (TYPE=XOT remote addressed by definition string - see R1CNOT in above sample) then all PVCs should activate in the first minute that HNAS is running (OPTIONS=PVCSETPTMR=secs on the TYPE=XOT REMOTE may be used to shorten this interval). If HNAS does not send SETUP packets then PVC session start depends on setup packet retry times in the router. Because the HNAS VTAM ACB is closed until the setup process completes it is not possible for the PLU to acquire the PVC LUs until the setup process completes. The VTAM state for a PVC LU will be CONCT (assuming that the HNAS application major node is active) while the setup process is in progress. The state will become ACTIV when the setup exchange completes.

PVC Setup Considerations

While the PVC Setup Status Codes section of the HNAS Messages and Codes guide provides a description of the status codes, the following information provides additional operational processes and difference between the Cisco router and HNAS PVC XOT services.

A PVC Setup will be sent after the initiator has successfully opened a TCPIP socket with the remote XOT service. If the remote TCPIP socket is not available the open request will time out. An XOT PVC session is considered connected once the PVC Setup exchange is successful. This does not imply that there is a session with the host PLU which is another process (see PVC Operational Notes above).

PVC XOT IPADDR Validation

HNAS does not force validation of the origination ipaddr for router initiating the PVC Setup request. As long as the PVC Setup contains valid PVC Setup information PVC initialization will be attempted.

CISCO, when the router accepts a socket open request from HNAS as part of the PVC Setup process, if the originating ipaddr from HNAS does not match the ipaddr on the Cisco router serial interface PVC string the PVC Setup will be rejected by the router. HNAS reports the rejection with a NAS7704W alert with STATUS=1B.

PVC XOT Setup Window and Packet Size Initialization

HNAS will accept and set the packet and windows size to values specified in the Cisco PVC Setup request.

CISCO will reject the HNAS PVC Setup request if the packet and windows size don't match the router PVC value.

PVC XOT Setup Initiation

HNAS, when configured via the TYPE=MCH REMOTE PVC= string */rmtname* option, will initiate outbound PVC Setup requests to the remote router. The frequency of SETUP attempts is controlled by OPTIONS=PVCSETUPTMR=secs (introduced under APAR 2400059, default secs=60). If a PVC Setup request arrives from the Cisco router before the HNAS timer processing completes then the router's PVC Setup will be utilized.

CISCO will initiate a PVC Setup when the router is activated (ios load) or after a PVC resource is added to a serial interface regardless of the state of the serial interface. The router will retry PVC Setups every 4 minutes when the HNAS destination in unavailable (not active or no path) or after a TCPIP Socket Close condition. There is current no way to configure the Cisco router to not send PVC Setups. The Cisco router will stop sending PVC Setups when a PVC Setup status code of x'10' or above is received. **Note:** When dynamically reconfiguring or removing Cisco router PVC definitions please be sure and issue '**no x25 pvc *n***' for each PVC that you plan to remove from the configuration prior to changing the **'x25 ltc *n***' value. This will ensure that no transient PVC definitions are still in the active configuration '**show x25 xot**'.

PVC XOT Setup to an already setup session

HNAS, when a PVC Setup is received for an already setup (initialized/connected PVC) HNAS will close the active PVC session and process the PVC Setup for the new session.

CISCO will reject an inbound PVC Setup request to an already setup (initialized/connected PVC) session.

This condition can result due to a configuration error. While Cisco router-to-router XOT PVC sessions are very static, HNAS permits this connect-disconnect-connect condition to provide greater flexibility in the configuration process and remote router switching capabilities.

PVC XOT Setup Termination

HNAS PVC's are closed when HNAS is shutdown (cancel, quit or abend) and when the HNAS console subsystem VARY ID=*pvc-pid* FORCE command is issued which causes the socket to be closed.

CISCO PVC's are closed when the router is shutdown or rebooted.

If a TCPIP Keepalive failure or a HNAS TAP=*value* Contact Lost condition occurs the PVC session will be closed (socket torn down) and standard PVC Setup reconnect attempts will be made as described above.

PVC CISCO Router Interface Outage and Shutdown Considerations

When a Cisco remote router tcpip ethernet or loopback interface goes down (network outage, is unplugged or shutdown), the XOT resources associated with the interface (such as x.25 serial interface connections) won't be disconnected until input arrives from either direction causing session keepalive processing time-outs.

When a Cisco remote router serial interface goes down (network outage, is unplugged or shutdown), the sockets for the XOT resources associated with the serial interface will be closed and HNAS will be informed of the outage immediately.

PVC HNAS Remote TYPE=XOT Outage and Shutdown Considerations

When a HNAS TYPE=XOT REMOTE is varied offline with the force option (VARY rmt-name OFFLINE FORCE) all resources associated with the remote will be disconnected and unavailable until the remote is reactivated. The VTAM resources are unbound with appropriate sense or status codes and the xot sessions will be cleared, reset and sockets closed. This causes the PVC XOT session to be closed (PVC Setup no longer active, disconnected). Router initiated PVC Setups will be rejected while the remote in offline.

When a HNAS TAP Contact Lost condition occurs, all resources associated with the HNAS TYPE=XOT REMOTE will be disconnected as described above (varied offline with the force option). When router Contact Reacquired occurs the resources will be available again. The new TAP OPTIONS=NOCLOSEONTAPFAILURE (introduced under APAR 2400055) will allow tapping functions to be performed although the remote won't be taken offline.

XOT HNAS-to-HNAS SVC Resource Definitions

HNAS provides the ability to communicate with another HNAS using a native IP backbone network without any X.25 facilities and their associated routers being involved. When two HNAS communicate with each other, one must take the role of connection initiator and the other of connection acceptor/responder. Each HNAS sees the other as a logical Cisco router. Since each HNAS uses the Cisco XOT transport protocol to envelop its data, the data enveloped at one end of the connection is readily extracted by the other end just as if it came from a Cisco router. The following example shows how SVCs can be used to established LU sessions between two HNAS images.

Initiator HNAS (INITHNAS) using PCNE (LLC0) LUs

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE GLOBAL PARAMETERS FOR INITIATOR HNAS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         BUILD NASNAME=INITHNAS,     ; SUPPLY INITIATOR PARMS
               :

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITLCL  LOCAL TYPE=XOT              ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4    ; INITHNAS IP ADDRESS
               PORT=1998             ; INITHNAS PORT NUMBER
               RTEOUT=(INITRMT)      ; ROUTE OUTBOUND CONNECT TO INITRMT
               RTEIN=(INITMCH)       ; ROUTE INBOUND TAP TO INITMCH

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CALLOUT RESOURCES
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITRMT  REMOTE TYPE=XOT             ; IDENTIFY XOT CLIENT
                IPADDR=192.40.60.5   ; RESPHNAS IP ADDRESS
                PORT=1998            ; INITHNAS INITIATES CONNECT
                VCLMT=4              ; CONNECTIONS ALLOWED
                TAP=0                ; NO TAPPING INITIALLY

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE LOGICAL MCH LINK FOR PCNE SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITMCH  REMOTE TYPE=MCH             ; IDENTIFY XOT MCH
                SVC0=(4,             ; 4 LLC0 SLUS FOR CALLOUT SVCS
                      INITLU01/310600010,
                      INITLU02/310600020,
                      INITLU03/310600030,
                      INITLU04/310600040)
                FAC=(420A0A          ; PKTSZ=1024
                     430707)         ; WINSZ=7
                :
```

Responder HNAS (RESPHNAS) using PCNE (LLC0) LUs

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE GLOBAL PARAMETERS FOR RESPONDER HNAS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         BUILD NASNAME=RESPHNAS,      ; SUPPLY RESPONDER PARMS
               :

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RESPLCL  LOCAL TYPE=XOT               ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.5     ; RESPHNAS IP ADDRESS
               PORT=1998              ; RESPHNAS PORT NUMBER
               RTEIN=(RESPMCH)        ; ROUTE INBOUND CONNECT TO RESPMCH

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CALLIN RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RESPRMT  REMOTE TYPE=XOT              ; IDENTIFY XOT CLIENT
                IPADDR=192.40.60.4    ; INITHNAS IP ADDRESS
                PORT=1998             ; RESPHNAS PORT NUMBER
                VCLMT=4               ; CONNECTIONS ALLOWED
                TAP=0                 ; NO TAPPING INITIALLY

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE LOGICAL MCH LINK FOR PCNE SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RESPMCH  REMOTE TYPE=MCH              ; IDENTIFY XOT MCH
                SVC0=(4,              ; 4 LLC0 SLUS FOR CALLIN SVCS
                      RESPLU01/31060001I0, ; <- APPLID=0 => CICS
                      RESPLU02/31060002I0,
                      RESPLU03/31060003I0,
                      RESPLU04/31060004I0)
                APPLNAME=(CICS)       ; LOGAPPL=CICS
                :
```

As is the case when HNAS services real remote users via a Cisco router serial interface, the TYPE=XOT LOCAL definition statement in each HNAS identifies the HOME IP address and supplies routing information for outbound connections (RTEOUT= for INITLCL) and inbound connections (RTEIN= for RESPLCL).

The TYPE=XOT REMOTE definition statement in each HNAS represents the other HNAS as a logical router. INITRMT in INITHNAS identifies the target IP address for outbound connections which is RESPLCL in RESPHNAS. RESPRMT in RESPHNAS identifies the source IP address for inbound connections which is INITLCL in INITHNAS. In other words, the IPADDR= operand values for the LOCAL and REMOTE definitions in INITHNAS must be exactly reversed in RESPHNAS. The REMOTE statements above also create the control blocks required to manage 4 XOT sessions across the IP backbone network.

Note: During our testing it was observed that when INITHNAS and RESPHNAS were connected to the same TCPIP stack, the source IP address from INITHNAS (from INITLCL) was replaced with the target IP address (from INITRMT). In this case, the IPADDR= value for RESPRMT and RESPLCL had to be set to the same value.

The TYPE=MCH REMOTE definition statement in each HNAS define a logical MCH. Note that the LUs in the SVC0= operands have a DTE address associated with them. For MCHINIT, they are *callout* DTE addresses. For RESPMCH, they are *callin* DTE addresses. For INITMCH, calls are placed to RESPMCH when the LUs identified in the SVC0= operand are bound. The DTE address shown above will guarantee that INITLU0*i* will always connect to RESPLU0*i*.

Each HNAS sees the other as a remote router. INITHNAS is the initiator of XOT socket connections and RESPHNAS is the acceptor of XOT socket connections. No real routers or X.25 facilities are required since each HNAS operates at the XOT level of the network protocol. Standard HNAS LLC0 *callout* and *callin* logic is employed to service the LU connection requests and data transfer. For more information on LLC0 SVC *callout* and *callin*, see text starting on page 3-9 and page 3-12, respectively.

General notes for HNAS-to-HNAS SVC support:

1) The IPADDR= value for the TYPE=XOT LOCAL in the initiator and responder HNAS (INITLCL and RESPLCL) represent the HOME IP address for each HNAS.

2) The IPADDR= value for the TYPE=XOT REMOTE in the initiator and responder HNAS (INITRMT and RESPRMT) must be set to the same value as the IPADDR= for the TYPE=LOCAL in the companion HNAS. A dynamic socket pool could also be used in RESPHNAS (IPADDR=DYNAMIC) if RESPRMT is also to be used to accept connections from other DTEs.

3) You may want to enable TAPping for each HNAS so that operations personnel will be able to tell whether the companion HNAS is idle or active. Please refer to the XOT HNAS-to-HNAS TAPping Considerations section below for additional information.

4) The DTE address values in the SVC0= operand for INITMCH and RESPMCH should be the same except that they represent outbound DTE addresses for INITMCH and inbound DTE addresses for RESPMCH. This mapping ensures that the LUs in INITHNAS will always connect to the same LUs in RESPHNAS.

5) Since no real X.25 network is involved in an HNAS-to-HNAS scenario, we recommend setting packet and window sizes to their optimum values in order to improve throughput and reduce HNAS CPU utilization. This is accomplished by coding the appropriate facilities values the FAC= operand (see INITMCH in the example above).

XOT HNAS-to-HNAS PVC Resource Definitions

HNAS provides the ability to communicate with another HNAS using a native IP backbone network without any X.25 facilities and their associated routers being involved. When two HNAS communicate with each other, one must take the role of connection initiator and the other of connection acceptor/responder. Each HNAS sees the other as a logical Cisco router. Since each HNAS uses the Cisco XOT transport protocol to envelop its data, the data enveloped at one end of the connection is readily extracted by the other end just as if it came from a Cisco router. The following example shows how PVCs can be used to established LU sessions between two HNAS images.

Initiator HNAS (INITHNAS) using PCNE (LLC0) LUs

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE GLOBAL PARAMETERS FOR INITIATOR HNAS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         BUILD NASNAME=INITHNAS,      ; SUPPLY INITIATOR PARMS
               :

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITLCL  LOCAL TYPE=XOT              ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.4    ; INITHNAS IP ADDRESS
               PORT=1998             ; INITHNAS PORT NUMBER
               RTEOUT=(INITRMT)      ; ROUTE OUTBOUND CONNECT TO INITRMT
               RTEIN=(INITMCH)       ; ROUTE INBOUND TAP TO INITMCH

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CALLOUT RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITRMT  REMOTE TYPE=XOT             ; IDENTIFY XOT CLIENT
                IPADDR=192.40.60.5   ; RESPHNAS IP ADDRESS
                PORT=1998            ; INITHNAS INITIATES CONNECT
                VCLMT=4              ; CONNECTIONS ALLOWED
                TAP=0                ; NO TAPPING INITIALLY

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE LOGICAL MCH LINK FOR PCNE SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
INITMCH  REMOTE TYPE=MCH             ; IDENTIFY XOT MCH
                PVC=(4,              ; 4 LLC0 SLUS FOR CONNECT OUT PVCS
                      INITLU01/0/0/01/SERIALRESPMCH/INITRMT/INITMXT,
                      INITLU02/0/0/02/SERIALRESPMCH/INITRMT/INITMXT,
                      INITLU03/0/0/03/SERIALRESPMCH/INITRMT/INITMXT,
                      INITLU04/0/0/04/SERIALRESPMCH/INITRMT/INITMXT)
                :

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE MCH EXTENSION TO SET PVC WINDOW AND PACKET SIZE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
INITMXT  REMOTE TYPE=MXT              ; IDENTIFY MXT
                FAC=(420A0A           ; PKTSZ=1024
                     430707)          ; WINSZ=7
```

Responder HNAS (RESPHNAS) using PCNE (LLC0) LUs

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE GLOBAL PARAMETERS FOR RESPONDER HNAS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         BUILD NASNAME=RESPHNAS,      ; SUPPLY RESPONDER PARMS
               :

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RESPLCL  LOCAL TYPE=XOT               ; IDENTIFY XOT SERVER
               IPADDR=192.40.60.5     ; RESPHNAS IP ADDRESS
               PORT=1998              ; RESPHNAS PORT NUMBER
               RTEIN=(RESPMCH)        ; ROUTE INBOUND CONNECT TO RESPMCH

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE CALLIN RESOURCES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RESPRMT  REMOTE TYPE=XOT              ; IDENTIFY XOT CLIENT
                IPADDR=192.40.60.4    ; RESPHNAS IP ADDRESS
                PORT=1998             ; RESPHNAS PORT NUMBER
                VCLMT=4               ; CONNECTIONS ALLOWED
                TAP=0                 ; NO TAPPING INITIALLY

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE LOGICAL MCH LINK FOR PCNE SESSIONS
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
RESPMCH  REMOTE TYPE=MCH              ; IDENTIFY XOT MCH
                PVC=(4,               ; 4 LLC0 SLUS FOR CONNECT IN PVCS
                      RESPLU01/0/0/01/SERIALINITMCH, ; <- APPLID=0 => CICS
                      RESPLU02/0/0/02/SERIALINITMCH,
                      RESPLU03/0/0/03/SERIALINITMCH,
                      RESPLU04/0/0/04/SERIALINITMCH)
                APPLNAME=(CICS)       ; LOGAPPL=CICS
                :
```

As is the case when HNAS services real remote users via a Cisco router serial interface, the TYPE=XOT LOCAL definition statement in each HNAS identifies the HOME IP address and supplies routing information for outbound connections (RTEOUT= for INITLCL) and inbound connections (RTEIN= for RESPLCL).

The TYPE=XOT REMOTE definition statement in each HNAS represents the other HNAS as a logical router. INITRMT in INITHNAS identifies the target IP address for outbound connections which is RESPLCL in RESPHNAS. RESPRMT in RESPHNAS identifies the source IP address for inbound connections which is INITLCL in INITHNAS. In other words, the IPADDR= operand values for the LOCAL and REMOTE definitions in INITHNAS must be exactly reversed in RESPHNAS. The REMOTE statements above also create the control blocks required to manage 4 XOT sessions across the IP backbone network.

Note: During our testing it was observed that when INITHNAS and RESPHNAS were connected to the same TCPIP stack, the source IP address from INITHNAS (from INITLCL) was replaced with the target IP address (from INITRMT). In this case, the IPADDR= value for RESPRMT and RESPLCL had to be set to the same value.

The TYPE=MCH REMOTE definition statement in each HNAS define a logical MCH. Note that the LUs in the PVC= operands have a logical channel number (LCN) associated with them. The LCN shown above will guarantee that INITLU0*i* will always connect to RESPLU0*i*.

Each HNAS sees the other as a remote router. INITHNAS is the initiator of XOT socket connections and RESPHNAS is the acceptor of XOT socket connections. No real routers or X.25 facilities are required since each HNAS operates at the XOT level of the network protocol. Standard HNAS LLC0 PVC logic is employed to service the LU connection requests and data transfer. For more information on LLC0 PVC connectivity, see text starting on page 3-30.

General notes for HNAS-to-HNAS PVC support:

1) The IPADDR= value for the TYPE=XOT LOCAL in the initiator and responder HNAS (INITLCL and RESPLCL) represent the HOME IP address for each HNAS.

2) The IPADDR= value for the TYPE=XOT REMOTE in the initiator and responder HNAS (INITRMT and RESPRMT) must be set to the same value as the IPADDR= for the TYPE=LOCAL in the companion HNAS. A dynamic socket pool could also be used in RESPHNAS (IPADDR=DYNAMIC) if RESPRMT is also to be used to accept connections from other DTEs.

3) You may want to enable TAPping for each HNAS so that operations personnel will be able to tell whether the companion HNAS is idle or active. Please refer to the XOT HNAS-to-HNAS TAPping Considerations section below for additional information.

4) The *rmtname* value (INITRMT) for the PVC= operand entries on INITMCH in INITHNAS force INITHNAS to send PVC Setup packets. The *rmtname* suboperand must be removed from the PVC= operand entries on RESPMCH in RESPHNAS so that it will not also send PVC Setup packets. It will only respond to them.

5) The 'serial interface' name in each PVC= operand entry must be coded as SERIAL*mchname* where *mchname* is the MCH name in the companion HNAS (if a real router were involved, this would probably look like SERIAL0-1). For INITMCH, *mchname* is RESPMCH. For RESPMCH, *mchname* is INITMCH.

6) The LCN values in the PVC= operand for INITMCH and RESPMCH should be the same. This mapping ensures that the LUs in INITHNAS will always connect to the same LUs in RESPHNAS.

7) Since no real X.25 network is involved in an HNAS-to-HNAS scenario, we recommend setting packet and window sizes to their optimum values in order to improve throughput and reduce HNAS CPU utilization. This is accomplished by associating an MXT with each PVC entry and coding the appropriate facilities values the FAC= operand (see INITMXT in the example above).

XOT HNAS-to-HNAS TAPping Considerations

You may want to enable TAPping for each HNAS so that operations personnel will be able to tell whether the companion HNAS is idle or active. The XOT TAP process uses a Call Request packet as a TAP request. The expected TAP response is a Clear Request packet. A failure to respond to a TAP request will alert the other HNAS that it's companion is down. This will result in the following messages being generated:

```
NAS2271W CLIENT=192.040.060.005(01998) SOCKID=0001 PCEID=0008 NAME=INITRMT
NAS2271W CONNECT REQUEST FAILED, RC=FFFFFFFF 0000003D
NAS2501W CLIENT=192.040.060.005(01998) SOCKID=0001 PCEID=0008 NAME=INITRMT
NAS2501W ROUTER KEEP ALIVE FAILURE 01 OF 02
NAS2271W CLIENT=192.040.060.005(01998) SOCKID=0001 PCEID=0008 NAME=INITRMT
NAS2271W CONNECT REQUEST FAILED, RC=FFFFFFFF 0000003D
NAS2501W CLIENT=192.040.060.005(01998) SOCKID=0001 PCEID=0008 NAME=INITRMT
NAS2501W ROUTER KEEP ALIVE FAILURE 02 OF 02
NAS2502E CLIENT=192.040.060.005(01998) SOCKID=0001 PCEID=0008 NAME=INITRMT
NAS2502E ROUTER CONTACT LOST
```

When the companion HNAS comes active, contact between the 2 HNAS images will be restored. This will result in the following messages being generated:

```
NAS2503W CLIENT=192.040.060.005(03057) SOCKID=0001 PCEID=0008 NAME=INITRMT
NAS2503W ROUTER CONTACT REACQUIRED
```

When both HNAS are active, TAPping between the 2 HNAS images should prevent each form loosing contact with the other. During this normal TAPping process, the following messages will be generated:

```
NAS2260I SERVER=192.040.060.004(01998) SOCKID=0000 PCEID=0007 NAME=INITLCL
NAS2260I CLIENT=192.040.060.005(03058) SOCKID=001F PCEID=0013 NAME=INITRMT
NAS2260I REMOTE CONNECTION ACCEPTED, SOCCNT=00002 SOCLMT=02000
NAS2210I SERVER=192.040.060.004(01998) SOCKID=001F PCEID=0007 NAME=INITLCL
NAS2210I SOCKET CONNECTION CLOSED, SOCCNT=00002 SOCLMT=02000
NAS2268I CLIENT=192.040.060.005(03058) SOCKID=001E PCEID=0013 NAME=INITRMT
NAS2268I ACCEPTED CONNECTION PASSED, SOCCNT=00002 SOCLMT=02000
NAS7715W 192.040.060.005(03058) CALL REQ TO MCH INITMCH FAILED, CLEAR DIAG=128
NAS2210I CLIENT=192.040.060.005(03058) SOCKID=001E PCEID=0013 NAME=INITRMT
NAS2210I SOCKET CONNECTION CLOSED, SOCCNT=00001 SOCLMT=02000
```

Note: The messages above are from the viewpoint of INITHNAS. Similar messages will be generated for RESPHNAS with the SERVER and CLIENT IP address reversed and the NAME= values starting with RESP instead of INIT.

Note: If you wish to suppress the informational TCPIP messages (NAS2*xxx*I), specify PRNTTCP OFF as a start parameter value, enter the PRNT TCP OFF console command or code ALRMFLTR=(A,NAS2***I(P),...) on the BUILD definition statement. You can also filter out all but the NAS2502E, NAS2503W and NAS7715W using the ALRMFLTR= operand.

We recommend specifying TAP=600 for INITRMT and TAP=660 for RESPRMT. This will minimize CPU utilization while still providing necessary companion status

information. The non-equal TAP= values also reduce the likelihood of a TAP collision.

Note: The RTEIN= operand on INITLCL in INITHNAS is only used to handle TAP Call Request packets from RESPHNAS. The RTEIN= operand on RESPLCL in RESPHNAS is used to handle real connections as well as TAP Call Request packets from INITHNAS. TAP Call Request packets form either HNAS will result in a Clear Request from the companion HNAS which will satisfy the TAP request/response criteria. If one of the HNAS images is down, the required response will not be received which will result in the NAS2502E message shown above.

XOT Datafono Resource Definitions

Datafono Introduction

HNAS Datafono support allows PLUs to communicate with Datafono devices using HNAS to provide the virtual SLU support formerly provided by NPSI VIRTUAL=YES resources. Datafono devices are used in Spain as point of sale terminals.

On the inbound flow the Datafono device communicates with an X25 PAD attached to a Cisco router via an X25 link. XOT support in the router forwards the X25 packets in TCP/IP datagrams over a WAN. HNAS receives the datagrams, removes the X25 control information, translates the data to EBCDIC and forwards the data in FMD PIUs using a VTAM application to application session (one session per active Datafono device). In the APPL-to-APPL session HNAS is the SLU.

On the outbound flow the above is reversed. HNAS is receives PIUs from the PLU. Data is packetized in XOT datagrams carrying X25 packets. The router receives the XOT datagrams and generates X25 packets which are sent via an X25 link to the Datafono PAD. The PAD forwards the data to the remote device.

The X25 protocol (processing of call request packets, X25 window rotations, etc.) is provided by HNAS.

Because HNAS uses APPL-to-APPL VTAM sessions to exchange data with the PLU there are some differences in the configuration process. The HNAS SLUs are defined to VTAM in an Application Major Node. There are no Switched Major Node’s associated with HNAS. The Application Major Node is normally generated by HNAS (Start parameter FASTRUN option). The customer is responsible for placing the node in a VTAM data set and activating the node. HNAS has facilities for defining the switched and leased resources expected by the PLU.

Please refer to the HNAS CDF Datafono Operands section of this guide for information regarding new Datafono operands for the TYPE=MCH REMOTE as well as the new TYPE= DFL, DFS and DFX REMOTE types resources.

Datafono Data Translation

Translation between ASCII (terminal code) and EBCDIC (host code) is made by using translate tables in the XAICDTFT CSECT. These tables were provided by IBM Spain. Some installations have custom versions of this table. In order to install a local version of XAICDTFT perform the following steps (APAR 2400011 must be applied):

Non-SMP/E installations:

Copy your version of XAICDTFT to hlq.HNASOBJX(XAICDTFT) and then run the HNASMNT job to rebuild the HNAS load module in hlq.HNASLOAD. To remove the custom translate table delete the hlq.HNASOBJX XAICDTFT member and rerun HNASMNT (the Comm-Pro version of XAICDTFT will still be in hlq.HNASOBJ).

SMP/E installations:

After the TSORECV job has been run the rrrr.LNS0240.F4 relfile PDS has the HNAS object members used to build HNAS. Preserve the LNSCDTFT member and then install your custom XAICDTFT as member LNSCDTFT in rrrr.LNS0240.F4 (relfile member names must start with the LNS prefix (assigned to Comm-Pro by IBM)). Your custom XAICDTFT can also be installed as a USERMOD. If this is done the MCS ++MOD statement must use the LNSCDTFT name.

The XAICDTF CSECT contains a table (ENTRY XDTFTABN) used to validate a REQUEST INITIALIZATION message from a terminal and to determine the number of times the EMSGE time-out message is sent by HNAS. HNAS does not validate the REQ INIT message and does not limit the number of times that the EMSGE message is sent.

Sample CDF Configuration File for Datafono Resource

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        HNAS V2R3M0 - SAMPLE DATAFONO CONFIGURATION.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         BUILD ,                     ;
               BFRLMT=500            ; BUFFER POOL REQUIREMENT.
               CONCMDQ=(DNAS,        ; START COMMANDS.
                        'TRCMCH ICR')
               CONLMT=2              ; NUMBER OF REMOTE CONSOLES
               CONPRMT='ZDTFCS1> '   ; CONSOLE PROMPT.
               CONPSWD=PASS          ; CONSOLE PASSWORD
               NASNAME=NAST0TSO      ; OUR NAME
               PRTLMT=2250000        ; TOTAL NUMBER PRINT RECORDS
               TCPNAME=TCPIP         ; TCP/IP ADDRESS SPACE NAME
               TRCLMT=4000           ; INTERNAL TRACE TABLE SIZE
               USSTAB=ISTINCDT       ; IBM STANDARD USSTAB
               VCLMT=100             ; SYSTEM WIDE VC LIMIT.
*
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOCAL XOT HOST PCSERVER/P390 - ETHERNET
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
LXOT     LOCAL TYPE=XOT                 ; TYPE
               IPADDR=10.117.56.170     ; LOCAL IP ADDRESS
               PORT=1998                ; XOT PORT NUMBER
               RTEIN=(MCH1)             ; ALL CALLS GO TO MCH1
*              RTEIN=(MCH1/2300,MCH2/7700,MCH3),
*                                       ; FOLLOWING WOULD ROUTE
*                                       ; CALLS TO AN MCH BY
*                                       ; BY CALLED DTE ADDR
                 RTEOUT=NONE            ; NO CALLOUT
               TCPNAME=TCPIP            ; TCPIP IS DEFAULT.
*
*
```

```
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*       REMOTE XOT - CISCO ROUTER PORTS (ROUTER INITIATED CONNECT)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1CNIN   REMOTE TYPE=XOT                ; ROUTER TRANSPORT PROTOCOL
               HOME=LXOT                ; THIS DEFAULT IS LXOT.
               INIT=ACTIVE              ; ACTIVATE AT STARTUP
               IPADDR=10.117.56.100     ; ROUTER IP ADDRESS
               PORT=DYNAMIC             ; ROUTER INITS CONNECT
               VCLMT=50                 ; ROUTER VC COUNT
*
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*       DEFINE LOGICAL MCH.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH            ; REMOTE TYPE
               APPLNAME=(TSO)       ; PLU FOR LLC5 SESSIONS
               CONNECT=NO           ; NO FAST CONNECT
               CTCP=(100,101,102,103,104); CUD0 TO LLC/CTCP MAP.
                                    ; A CTCP VALUE OF 100 THRU 120
                                    ; INDICATES A DATAFONO CALL WITH
                                    ; ATTRIBUTES SUPPLIED BY A TYPE=DFX
                                    ; REMOTE ADDRESSED BY THE DFXNAME=
                                    ; PARAMETER (BELOW).  IF THE CTCP
                                    ; NUMBER IS 100 THE FIRST DFXNAME=
                                    ; NAME IS USED.  101 = SECOND NAME,
                                    ; 120 = 21st NAME.
               CUD0=(20,21,22,23,24); CUD0 VALUES.
                                    ; ABOVE CUD0 VALUES ESTABLISH
                                    ; A DATAFONO CALL BECAUSE THE
                                    ; CORRESPONDING CTCP SLOT IS
                                    ; IN THE RANGE 100-120.
                                    ; STD NPSI VALUES STILL APPLY
                                    ; (E.G. 01=LLC5).
               DFLNAME=(DFL001)     ; LEASED LU POOL. UP TO 15
                                    ; POOLS MAY BE ADDRESSED.
                                    ; EACH POOL PROVIDES UP TO 64
                                    ; LEASED LU NAMES USED WHEN XID=NO IS
                                    ; CODED ON A TYPE=DFX REMOTE.
                                    ; SEE TYPE=DFL REMOTE, BELOW.
                        DFXNAME=(DFX001,; LEASED DFX.
                        DFX002,     ; XID=STD DFX.
                        DFX003,     ; XID=(NNNNN,MM) DFX.
                        DFX004,     ; XID=TAB,NRITAB=ZDTFT01 DFX.
                        DFX005)     ; NO OPTIONS = XID=STD
                                    ; SEE TYPE=DFX REMOTES, BELOW.
               GATE=GENERAL         ; GATE SUPPORT
               IDLETO=0             ; NO INACTIVITY TIMER
               LLC0=NONE
               LLC4=NONE
               LLC5=(5)             ; 5 SETS LLC5
               MBITCHN=YES          ; BUILD M-BIT CHAINS
               PAD=INTEG            ; IPAD
```

```
                PADPARM=(13/7)          ; INSERT LF'S
                PVC=NONE                ; NO PVC SUPPORT.
                SUBADDRESS=YES          ; SUBADDRESS CAN SET LLC
                SVC0=NONE               ; NO LLC0
                SVC4=NONE               ; NO LLC4
                SVC5=(3,                ; CALL IN LU'S.
                      MCH15001,MCH15002,MCH15003)
                TRAN=SPACE              ; LLC5 TRANSLATE TABLE.
*
*
*        DFX REMOTES
*        SELECTED BY CUD0 VALUE IN CONJUNCTION WITH CTCP=
*        AND DFXNAME=
DFX001   REMOTE TYPE=DFX
                OPTIONS=(DATAF,EMSGE,RETPIU,XID=NO,IMS)
DFX002   REMOTE TYPE=DFX
                OPTIONS=(DATAFAM,XID=STD)
DFX003   REMOTE TYPE=DFX
                OPTIONS=(DATAFAM,XID=(77771,7))
DFX004   REMOTE TYPE=DFX
                OPTIONS=(DATAF,XID=TAB,NRITAB=ZDTFT01)
DFX005   REMOTE TYPE=DFX
                OPTIONS=DATAF
*
*
*        DFL (DATAFONO LEASED) REMOTES
*        WHEN AN MCH RECEIVES A DATAFONO CALL AND CUD0=, CTCP= &
*        DFXNAME= INDICATE XID=NO, THEN THE MCH'S DFLNAME= PARM
*        IS USED TO LOCATE A LEASED RESOURCE IN A TYPE=DFL REMOTE.
*        THE ENTRY FORMAT IS ...,SLU-NM/PLU-NM.
*        IF THE PLU NAME IS CODED HNAS ISSUES A REQSESS TO ASK FOR
*        A SESSION.  IF THERE IS NO PLU-NM THE PLU MUST ACQUIRE
*        THE HNAS SLU.
DFL001   REMOTE TYPE=DFL
                LUNAME=(MCH10001/NASCTCP,MCH10002)
*
*
*        DFS (DATAFONO SWITCHED) REMOTES
*        WHEN ANY MCH RECEIVES A DATAFONO CALL AND CUD0=, CTCP= &
*        DFXNAME= INDICATE XID=YES, THEN THE HNAS SWITCHED LU POOL
*        IS SEARCHED TO LOCATE A RESOURCE WITH THE CORRECT IDBLK.
*        EACH TYPE=DFS REMOTE CAN SPECIFY 64 LUS.  ANY NUMER OF
*        THESE MAY BE CODED.
*        THE ENTRY FORMAT IS ...,SLU-NM/PLU-NM/IDNUM,...
*        ALL 3 OPERANDS ARE REQUIRED.
DFS001   REMOTE TYPE=DFS
                LUNAME=(MCH10003/NASCTCP/77777,
                        MCH10004/NASCTCP/77778)
DFS002   REMOTE TYPE=DFS
                LUNAME=(MCH10005/NASCTCP/77779
                        MCH10006/NASCTCP/7777A)
*
*
         END
```

XTP Definition Summary

IBM routers using the XTP protocol use a single TCP session to carry all traffic to and from a router. HNAS needs to know the configuration of the router -- specifically which interfaces are connected to X25 lines. An XOT TCP segment sent to the router's IP address must carry a router interface number identifying the interface the data is to be sent on. Each segment received from the router also carries an interface number so HNAS knows which MCH the data is associated with.

The router is defined with definition statements describing the X25 configuration to be supported at each X25 router interface.

```
R1MCH2   REMOTE TYPE=XTP
                IPADDR=192.66.77.21  ; ROUTER IP ADDRESS
                PORT=3065            ; WELL KNOWN XTP PORT
                IFNUM=2              ; INTERFACE NUMBER FOR X25 MCH LINK
                                     ; BEING DEFINED
                SVC0=(...),SVC4=(...),SVC5=(...)
```

Statements like the above identify an XTP router (by IP address). IFNUM identifies an interface number on the router that connects to an X25 link. Following parameters describe the session types supported on the X25 link and the HNAS resources required by the various session types. Additional TYPE=XTP REMOTE statements with the same IP address and different IFNUMs are used to describe other X25 links attached to the same router.

The following pages contain a configuration supporting LLC0 (PCNE), LLC4 (GATE) and LLC5 (PAD) callin and callout sessions. Descriptions of how HNAS processes inbound and outbound calls follows the definition statements. The example provides an overview -- be sure to consult Chapter 4 for a description of all operands.

XTP PCNE, GATE and PAD Resource Definitions

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        DEFINE XTP ROUTER WITH X25 LINK
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
R1MCH1   REMOTE TYPE=XTP IPADDR=192.66.77.21 PORT=3065 IFNUM=2
                PVC=(1,LUP01,5,1)       ; DEFINE LLC5 PVC
                SVC0=(3,                ; DEFINE 3 SLUS FOR LLC0 SVCS
                      LU001,            ; CALLIN SLU NAME
                      LU002/X123450I0,  ; SELECTED BY IDNUM, PLU IS PLU1
                      LU003/30167770)   ; CALLOUT SLU WITH CALLED DTE ADDR
                GATE=GENERAL CUD0=ALL   ; ALLOW LLC4, STANDARD CUD0/LLC MAP
                SVC4=(2,LU4001,LU4002)  ; CALLIN/OUT GATE DATA SES SLUS
                LUNAME=(CTCP1,CTCP2)    ; GATE CONTROL SES SLUS
                SVC5=(3,                ; DEFINE 3 LLC5 SLUS
                      LU501,            ; CALLIN SLU NAME
                      LU502,            ; CALLIN SLU NAME
                      LU503/30167770)   ; CALLOUT SLU WITH CALLED DTE ADDR
                PAD=INTEG,              ; HNAS PROVIDE LLC5 PAD SUOPPORT
                TRAN=EVEN               ; LLC5 ASCII TRANSLATE TABLE
                SYSL=(SUBD=1/0,SUBD=09/1,SUBD=3/2)) ; MAP SUBADDRESS TO AN
                                        ; INDEX IN APPLNAME
                APPLNAME=(PLU1,CONSOLE,MCHSOL) PLU NAMES LLC0/5 SESSIONS
```

XTP Inbound Call Processing

When an XTP call is received the fist step is to determine the LLC. Since GATE=GENERAL and CUD0=ALL are specified the standard IBM GATE LLC selection table is used:

LLC0: C0, CC
LLC5: 01, 41, 51, 81
LLC4: 02, C4 with first CTCP
      00, 03-2F or NULL with second CTCP.

See Chapter 4 for information on supplying custom CUD0 mappings (CUD0=, CTCP=) and on LLC selection via subaddress (SUBADDR=. LLC0=, etc.).

When a call is received, HNAS sets the LLC based on the first call user data byte (CUD0). When the LLC is known an SLU control block is allocated from a queue of available callin SLUs. There is one available queue for LLC0, LLC4 and LLC5 resources created by the SVC0=, SVC4 and SVC5= operands.

XTP PCNE (LLC0) and PAD (LLC5) Callin

SLU control block allocation is a 2 step process. First the appropriate available queue is searched for an SLU configured with a hex digit string or DTE address matching the call request packet's IDNUM value in call user data bytes or the *calling* DTE address (SLUs generated without a hex digit string or DTE address are skipped in step 1). If a match is found the SLU control block is allocated (removed from the available queue, placed on the active queue). If there is no match, the available queue is searched again. On the second pass an SLU control block is allocated only if no hex digit string or DTE address was coded on the SVC0= or SVC5= operand. If no SLU control block can be located the call is cleared. In the above, SLUs LU001, LU501 and LU502 can be used by any inbound call with the correct LLC type. SLU LU002 can only be used by an LLC0 caller with CUD1-3=X'123450'.

After SLU allocation HNAS determines the PLU name that will be used for the session. There are 3 sources for the PLU name:

1) APPLNAME= index coded with the SLU name. When the second component in the SLU's definition in the SVC0= or SVC5= parameter ends with Id (Inbound, decimal digits) then the SLU is reserved for inbound calls and the PLU is selected by using the digit(s) following the I as a select index in APPLNAME=. In the above LU002 has 'I0' following the hex digit string (remote's IDBLK). The '0' (following the 'I') used as an index in APPLNAME=, produces the PLU name of PLU1 (this PLU is always used by LU001).

2) If the SLU definition does not include an APPLNAME= index then the SYSL= parameter is used to produce an APPLNAME index from subaddress digit(s) or a call user data byte. If SYSL= is omitted or fails to produce an index then the first APPLNAME= entry is selected.

3) If 1) or 2) selects MCHSOL (reserved word in APPLNAME=) then the PLU name is selected by data from the remote in conjunction with an interpret table (LOGTAB=, not used above) or a USS Table (USSTAB=ISTINCDT, is the default IBM table).

In the above, the definition strings for LU001, LU501 and LU502 do not include an APPLNAME= index. The SYSL= is used to select an APPLNAME= entry based on subaddress digits (the last 2 digits in the call request packets *called* DTE address. In the above subaddress 1 selects PLU1, 09 selects the HNAS console services routine and 3 selects MCHSOL. The CONSOLE 'PLU' is a HNAS component that allows a remote to communicate directly with HNAS. The available commands are described in the "HNAS CONSOLE Subsystem and Operations Guide" MCHSOL directs HNAS to produce a PLU names using data from the remote processed against an interpret table (LOGTAB=, not used in this example) or against a USS Table. USS Messages, if defined, are sent. With the default ISTINCDT table the remote is expected to enter LOGON APPLID(xxx) to identify the PLU.

After the SLU and PLU names are known, HNAS sends a REQSESS to VTAM to request a session with the PLU. If the REQSESS triggers a BIND from the PLU, then PLU<->HNAS SLU<->remote device communication can proceed. If a BIND is not received the call is cleared.

When MCHSOL is used, HNAS sends a call accept so that the LOGON message can be entered. If the PLU name came from the APPLNAME= parameter then the call accept is sent when HNAS receives the PLU's BIND.

After the LLC0/LLC5 SLU and PLU names are known, HNAS sends a REQSESS to VTAM to request a session with the PLU. If the REQSESS triggers a BIND from the PLU, then PLU<->HNAS SLU<->remote device communication can proceed. If a BIND is not received the call is cleared.

XTP PCNE and PAD Callout

LU003 and LU503 are callout resources (indicated an ‘O’ following the DTE address). When a PLU acquires one of these resources HNAS receives a BIND. The BIND triggers HNAS to generate a call request packet with the *called* DTE address from the SVC0= or SVC5= operand. A *calling* DTE address and facilities may be supplied by DCEADDR= and FAC= keywords.

XTP GATE (LLC4) Callin and Callout

When CUD0- selects GATE (LLC4) an SLU is allocated from the LLC4 available queue created when initialization code processes the SVC4= operand. LLC4 SLUs can be used for input or output. After SLU allocation the remote’s call request is sent to a CTCP using one of the two control session SLUs - CTCP1 and CTCP2. CUD0=ALL indicates that the selection is based on CUD0 (see XTP Inbound Call Processing, above). When the CTCP returns a call accept HNAS forwards the packet to the remote, opens the data session ACB and sends a REQSESS to VTAM requesting a session for the data session LU. When the CTCP binds the data session SLU data can flow.

Some CTCPs expect the Resource ID carried in an incoming call request packet to identify a specific SLU. HNAS creates resource ids based on the position of the SLU name in the SVC4= operand offset by the number of PVCs defined. If there are no PVCs and If there are 3 SVC4= SLUs, the resource ids will be 1 through 3. If there are 2 PVCs defined, the resource ids will be 3 through 5. These considerations are important if the CTCP has a table mapping resource ids to SLU names.

For callout, the CTCP sends a call request packet to HNAS on the control session LU. This causes SLU allocation (from the LLC4 available queue) and generation of a call request packet. Called and calling addresses come from the CTCP. When the remote returns a call accept HNAS forwards it to the CTCP and sends a REQSESS to VTAM requesting a session for the data session SLU. When the CTCP binds the data session SLU data can flow.

CHAPTER 4

Configuration Statement Reference

This page left intentionally blank.

Chapter 4 - Configuration Statement Reference

This chapter describes the definition statements that are used to configure the Comm-Pro X.25 Host Network Access Support (HNAS) program product.

HNAS configuration definition statements are placed in a Configuration Data File (CDF) that is interpreted by the HNAS program when it activates.

Multiple copies of the HNAS software can be executing simultaneously in different address spaces on the same host.   A separate CDF is required for each HNAS address space.

The descriptions of individual definition statement operands identify the resource types for which the operands are valid. If no resource type is indicated, the operand is valid for all resource types.

Please refer to Chapter 2 section Configuring Host NAS for an overview of the configuration process as related to installation, generation and activation of the HNAS product.

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).

Definition Statement Coding Conventions

The descriptions of the definition statements in this manual use the following coding conventions.

- **an asterisk * or Euro not-equal symbol (X'5C')**
  as the first character of a CDF record indicates that the entire record is to be treated as a comment. This character is also used as a wildcard character within certain operands (see ALRMFLTR=, RTEIN=, RTEOUT= for examples).

- **a pound sign # or French/Italian £ or Spanish Ñ (X'7B')**
  as the first character of a CDF record indicates that the entire record is to be treated as a comment. Unlike the comment that starts with an asterisk, this comment is also displayed in the HNAS SYSPRINT file. In this way, you can create block comments that may be placed in front of definition statements or definition statement operands in the CDF that will be listed in SYSPRINT when the CDF is scanned.

- **a semicolon ; (X'7E')**
  placed anywhere on a CDF record, indicates the start of a comment. The semicolon and any data to its right is not parsed as an operand value.

- **a plus sign + (X'4E')**
  placed after all configuration data but before the semicolon that starts a comment on a CDF record indicates that the current configuration data is continued on the next CDF record. The first non-blank character on the CDF record identifies the start of data. The first non-blank character that precedes the plus sign identifies the end of data. You may continue configuration data in this way for up to 1024 characters. When a CDF record without a plus sign is decoded, it signals the end of the CDF record continuation at which time the collected data is processed. The plus sign, in effect, acts as a 'more data' indicator.

  For example:

```
SVC0=(5,
      PCNESLU1/ +           ; PCNE SLU #1
      /1036000301 +         ; first callout DTE address
      -1036000102 +         ; second callout DTE address
      -666612345333301O +   ; third callout DTE address
      /MXTSESBL +           ; MXT for SLU
      /C0000000,            ; callout CUD
      PCNESLU2,PCNESLU3, +  ; PCNE SLU #2 and #3
      PCNESLU4,PCNESLU5)    ; PCNE SLU #4 and #5
```

  will be processed (parsed) as follows:

```
SVC0=(5,
         PCNESLU1/1036000301-1036000102-666612345333301O/MXTSESBL/C0000000,
         PCNESLU2,PCNESLU3,PCNESLU4,PCNESLU5)
```

- **an at sign @ or French à or German/Italian § (X'7C')**
  as the first *dteaddr* character of a RTEIN= or RTEOUT= DTE address tells HNAS to accept any DTE address whose trailing digits match the *dteaddr* digits that follow the @. For exam-

ple, @6789 will allow DTE addresses of 123456789, 16789, 3216789, etc. to be considered a match.

- **CAPITALIZED KEYWORD TEXT**
  (like MCHSOL) represent values that you code as given, without change.

- **small *italicized* keyword text**
  (like *vclmt*) represent parameters for which you must supply a value.

- **small keyword text**
  (like tcbaddr) represent variable parameters that are computed by HNAS which cannot be changed.

- **parentheses ( )**
  are used to enclose a sequence of values that may be coded in one operand.

- **commas ,**
  are used to separate operands or a sequence of values that may be coded in a single operand.

  Note: A comma is required to separate multiple operand values that are specified on the same CDF record. A comma is optional following the last operand or suboperand of a sublist operand on a CDF record. For example, the following CDF records are both accepted:

  MCH1 REMOTE TYPE=MCH,SVC0=(20,

  - or -

  MCH1 REMOTE TYPE=MCH,SVC0=(20

- **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 (//).

- **brackets [ ]**
  enclose operands or symbols that are either <u>optional</u> or <u>conditional</u>.

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

  A conditional operand is one that you can code or omit, depending how you code or omit other operands on the same or higher level definition statements.

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

- **a 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.

- **an underlined value**
  indicates the default value for the operand. This is the value the HNAS configuration assumes if you omit the operand.

  If the underlined value is given as `*`, a default value appears in a column to the right of the operand.

  If the underlined value is given as `*+1`, the next unused value from a list of values is used as the default for the operand. All used values are remembered.

  If the underlined value is another operand name (e.g., **VCLMT**), the default value for the operand is taken from the named operand on the same definition statement.

  If the underlined value is a definition statement name (e.g., **BUILD**), the default value for the operand is taken from the same operand on the named definition statement.

  If the underlined value is another operand name preceded by + (e.g., **+VCMLT**), the default value for the operand is taken from the sum of the named operand on the lower level definition statements in the CDF.

- **ellipsis ... (3 consecutive dots**)
  indicates that you can code the preceding item more than once or, for a sublist, indicates that each entry has the same basic form which can be repeated (e.g., CUD0=(value1,...,valuen) or RTEIN=({*mchname*|CLEAR|SKIP}/*dteaddr*{S|T},...)).

- **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 parenthesis. For example, CTCP=(0,0,1,2,0).

- **sublist continuation statements**
  allow sublist suboperands to be specified on multiple CDF records. The first suboperand must immediately follow the opening parenthesis that starts the sublist. The last suboperand must immediately precede the closing parenthesis that ends the sublist. Intervening suboperands can appear on individual CDF records between the first and the last.

  For example, CTCP=(0,0,1,2,0) can be coded on multiple CDF records as follows:

```
CTCP=(0,
      0,
      1,2,
```

```
0)
```

Note: Plus sign (+) continuation is not required for sublist continuation unless an individual sublist suboperand and its associated comments exceeds the CDF record length of 80 characters.

- **numeric**
  values are always represented in decimal unless noted otherwise.

- **numeric conversion**
  always ignores leading zeroes unless noted otherwise. For example, CUD0=0C and CUD0=C are treated the same way. The value stored in memory is saved as a byte value of 0C.

- **indices**
  into list form operands are given relative to zero unless noted otherwise. For example, CTCP=(0,0,1,2,0) has 5 elements and is indexed from 0 to 4.

- **reserved words**
  may not be used to identify the names of PLUs, SLUs or MCHs (REMOTEs) unless noted otherwise. For example, CONSOLE, MCHSOL, SKIP and CLEAR are reserved by HNAS for special processing.

- **default values**
  are used when a parameter is omitted or specified incorrectly. Parameters specified without a value will be flagged by a warning message (RC=4) but their default will be used. If you want to eliminate the warning message and prevent use of the default, you must specify a valid value for the parameter.

- **error substitution values**
  of 0 or 255 are used when a subparameter is specified incorrectly (e.g., 90 specified as a CTCP= list entry, 85 is the maximum allowed). Invalid subparameter values will be flagged by a warning message (RC=4) but their substitution value will be used. If you want to eliminate the warning message and prevent the substitution, you must specify a valid value for the subparameter.

- **NULL|NONE values**
  NULL was added so that parameters on an MXT could be overridden by parameters of the same name on the root MCH when an MXT is associated with an SLU (see PVC=, SVCi= operands). NONE for an MXT parameter means that the parameter should be omitted regardless of what is specified for the same parameter on the root MCH. NULL|NONE are treated the same when specified on the root MCH. NULL is valid for the CUD=, FAC=, DTEADDR= and DCEADDR= operands only.

  NONE was initially implemented to allow users to eliminate default informational and/or warning messages to prevent cc-4 conditions for some assumed parameter defaults and is still supported in this manner.

- The LOCAL and REMOTE definition statements in this chapter appear in columnar form with parameter choices identified as being **Required (R)**, **Invalid (I)**, **Ignored (N)** or **Optional**

(blank). The table heading identifies the HNAS LOCAL server types (XTP and XOT) and REMOTE client types (XTP, XOT, MCH, MXT and SPU) and all of the parameter choices supported under each type. For Cisco routers TYPE=XOT|MCH|MXT|SPU are supported while for IBM routers TYPE=XTP|MXT parameters are supported.

- **CC/VRM heading entry values**
  are used to denote Change Control 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 (for the BUILD, LOCAL or REMOTE definition statement) where the operand is described. This will allow you to identify CC/VRM changes at the suboperand level.

Definition Statement Record Processing Rules

Before you begin coding the definition statements for the HNAS CDF, you should be aware of the way CDF records are processed by the HNAS configurator.

Each CDF record can be 80 characters in length (required if the CDF is a MACLIB member). For MACLIB members, the sequence number field (record columns 73-80) can also be used as part of the data as long as the **ISPF EDIT 'number off'** profile option is enabled. The CDF can also be stowed as a sequential file with an LRECL larger than 80 although we don't recommend it because records larger than 80 characters are truncated when they are read by the HNAS configurator.

Note: If you create your CDF using ISPF EDIT and the **'number on'** profile option is in effect, record columns 73-80 will contain a sequence number supplied by the editor. Since the CDF scanner processes each CDF record for 80 characters, the sequence number field can cause errors unless it is preceded by a semi-colon which identifies everything that follows as a comment. To avoid having to supply a semi-colon on every record that doesn't already have a comment, we recommend turning sequence numbering off by changing the 'number on' profile option to 'number off' (NONUM for short). This will prevent the ISPF editor from adding sequence numbers in record columns 73-80. These columns will remain blank so that they can be used for real CDF information.

Each CDF record can contain one operand or a number of operands per record as long as the coding rules are followed. Each operand must be separated by a comma.

When a semicolon is present, the CDF record data field ends there.

With the exception of the CUD= and FAC= operands on the REMOTE definition statement, only sublist operands (e.g., SVC0=) are allowed to be continued from one CDF record to another with each suboperand being separated by a comma. If all the sublist entries can fit on one CDF record, there is no need to code multiple records but you can if it enhances the readability. The number of continuation records for a sublist operand is a function of the size of the array that will hold the sublist information. When the array is full, continued sublist operand records will be rejected with the following message:

NAS1003W LIMIT: ***defname opname=opval***

You can code a + sign at the end of a CDF record to prevent the record from being parsed until a record without a + sign is decoded. All the data is concatenated into one text area of length 1024. The entire text area is then parsed as one single 'record' which can contain multiple operands, a single sublist suboperand or a complete sublist as is shown in the preceding section. Only non-blank characters are saved in the 1024 text area.

The reason we showed the SVC0= operand when describing the + sign continuation logic in the preceding section was to illustrate 2 types of continuation. + sign continuation is different than sublist continuation but both can be used together.

Definition Statement Placement

Before you begin coding the definition statements for the HNAS program, you should be aware of the order in which they must appear in the CDF. The placement of definition statements, as well as the requirements for their usage, is described in the following table.

STATEMENT	USAGE	POSITION
PARSE	Optional. This definition provides configuration parsing options that are extracted during the CDF pre-scan pass and subsequently used during the actual CDF scan.	Must be the **first** definition in the CDF.
BUILD	Required. This definition identifies global system parameters that are common to all resources in the HNAS address space.	Must be the **first** definition in the CDF if PARSE is not coded or the **second** if PARSE is coded.
LOCAL	Required. This definition identifies a server to the HNAS address space as it appears to the TCP/IP stack. At least one HOME LOCAL definition is required for each protocol server (e.g., XOT or XTP).	Can be placed anywhere between the BUILD and END definition statements.
REMOTE	Required. This definition identifies a router (TYPE=XOT|XTP) to the HNAS address space. At least one TYPE=XOT|XTP REMOTE definition is required for each static router and at least one REMOTE definition for dynamic router environments.	Can be placed anywhere between the BUILD and END definition statements.
	REMOTE types: Logical MCH (TYPE=MCH), MCH Extension (TYPE=MXT), QLLC SPU (TYPE=SPU), Datafono Leased SLU control (TYPE=DFL), Datafono Switched SLU control (TYPE=DFS), Datafono MCH Extension (TYPE=DFX), PING console command control (TYPE=DMY) or LLC0/5 callout control (TYPE=SVC) are also provided as a separate REMOTE definition.	-

STATEMENT	USAGE	POSITION
MCH	Optional (new for V2R2M0). This definition identifies a logical MCH to the HNAS address space. This statement is treated the same as REMOTE TYPE=MCH.	Can be placed anywhere between the BUILD and END definition statements.
MXT	Optional (new for V2R2M0). This definition identifies an MCH Extension to the HNAS address space. This statement is treated the same as REMOTE TYPE=MXT.	Can be placed anywhere between the BUILD and END definition statements.
SPU	Optional (new for V2R2M0). This definition identifies a QLLC SPU to the HNAS address space. This statement is treated the same as REMOTE TYPE=SPU.	Can be placed anywhere between the BUILD and END definition statements.
DFL	Optional (new for V2R4M0). This definition identifies a Datafono Switched SLU control definition to the HNAS address space. This statement is treated the same as REMOTE TYPE=DFS.	Can be placed anywhere between the BUILD and END definition statements.
DFS	Optional (new for V2R4M0). This definition identifies a Datafono Leased SLU control definition to the HNAS address space. This statement is treated the same as REMOTE TYPE=DFL.	Can be placed anywhere between the BUILD and END definition statements.
DFX	Optional (new for V2R4M0). This definition identifies a Datafono MCH Extension to the HNAS address space. This statement is treated the same as REMOTE TYPE=DFX.	Can be placed anywhere between the BUILD and END definition statements.
DMY	Optional (new for V2R4M0). This definition identifies a a dummy REMOTE that supplies parameters for the Console Subsystem PING command. This statement is treated the same as REMOTE TYPE=DMY.	Can be placed anywhere between the BUILD and END definition statements.

STATEMENT	USAGE	POSITION
SVC	Optional (new for V2R4M0). This definition identifies a REMOTE that supplies parameters for LLC0 and/or LLC5 callout SVCs. This statement is treated the same as REMOTE TYPE=SVC.	Can be placed anywhere between the BUILD and END definition statements.
END	Optional (recommended). This definition performs configuration cleanup functions which include validating parameters and setting defaults. If the statement is omitted, the end of data (EOD) condition signals the END function. **Note**: Any records that follow the END statement are ignored by the configuration process. This is a good location to place alternate or backup test definition statements.	Must be the **last** definition in the CDF.

Definition Statement Parameter Cross Reference

Operands	BUILD	LOCAL	REMOTE	ZOS	OS390	MVS	CC/VRM or Notes
APPLNAME=		I					C/230
ALRMCART=		I	I				230
ALRMFLTR=		I	I				210, C/220
ALRMLMTS=		I	I				220
BFRLMT=		I	I				
BFRSIZ=		I	I				C/240
CONCMDQ=		I	I				210, C/240
CONLMT=		I	I				
CONPRMT=		I	I				114
CONPSWD=		I	I				
CONNECT=	I	I	E				
CTCP=	I	I	E				
CUD=	I	I	S				113, C/220, 1)
CUD0=	I	I	E				
DCEADDR=	I	I					113, C/220
DFLNAME=	I	I					N/240
DFXNAME=	I	I					N/240
DTEADDR=	I	I					230
EXEC=		I	I				A/240
FAC=	I	I	S				113, C/220, 2)
GATE=	I	I	E				
HOME=	I	I					220
IDBLK=	I	I					220
IDNUM=	I	I					220
IDLETO=		I	B				114, C/230
IFNUM=	I	I					
INIT=	I						112, C/230
IPADDR=	I						C/210
LLC0=	I	I	E				
LLC3=	I	I	E				220
LLC4=	I	I	E				
LLC5=	I	I	E				
LOGTAB=		I	B				210, C/220

Operands	BUILD	LOCAL	REMOTE	ZOS	OS390	MVS	CC/VRM or Notes
LUNAME=	I	I	E				C/220
MAXDATA=	I	I					220
MBITCHN=	I	I	E				C/210
MSGLMT=		I	I			N	
NASNAME=		I	I				
OPTIONS=							210, C/211, C/220, C/230, C/240
PACE=	I	I					112, C/220
PAD=	I	I	E				
PADPARM=	I	I					
PKTSIZ=	I	I					C/220
PORT=	I						
PROTOCOL=	I	I					N/240
PRTLMT=		I	I				
PRTSWLST=		I	I				N/240
PULSE=		I	I				A/240
PVC=	I	I					
PWPROT=	I	I	E				
RTEIN=	I		I				112, C/210
RTEOUT=	I		I				112, C/210, C/220
SCHEDULE=		I	I				A/240
SOCLMT=	I		I				220
SUBADDR=	I	I	E				
SUBD=	I	I	E				
SVC0=	I	I					C/211, C/220
SVC3=	I	I					220
SVC4=	I	I					
SVC5=	I	I					C/211, C/220
SYSL=	I	I					C/211, C/220
TAP=	I	I					C/210, C/220
TCPNAME=		B	I				C/220
TRAN=	I	I	E				C/230
TRCLMT=		I	I				C/220
TRCTRAP=		I	I				A/230, C/240

Operands	BUILD	LOCAL	REMOTE	ZOS	OS390	MVS	CC/VRM or Notes
TYPE=	I						112, C/220
USSTAB=		I	B				114, C/220
VCLMT=		I					C/220

Figure 6: Definition Statement Operand Cross Reference

Legend for Figure 6 and the Definition Statements that follow

blank indicates that the operand is **valid** for the named statement
B indicates that the operand is **valid** for the named statement but defaults to the BUILD value when omitted
I indicates that the operand is **invalid** for the named statement or system
N indicates that the operand is **ignored** for the named statement or system
E indicates that the operand is **equivalent** to a NPSI operand
S indicates that the operand is **similar** to a NPSI operand
R indicates that the operand has been **retired**

CC/VRM for Figure 6

CC/VRM denote Change Control activity for the HNAS **V**ersion, **R**elease and **M**odification level where the changes were introduced. **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.

Notes for Figure 6

1. Provides a function similar to the USRFILD operand of the NPSI X25.OUFT statement.
2. Provides a function similar to the OPTFACL operand of the NPSI X25.OUFT statement.

PARSE Definition Statement

The PARSE definition statement is used to specify configuration parsing options. The parsing options are extracted during the CDF pre-scan pass and are then subsequently used during the actual CDF scan process. The PARSE definition statement is optional. If specified, it must be the first definition statement in the CDF (before BUILD). The PARSE definition statement allows customers to specify their own parsing characters that will override established HNAS parsing characters for the CDF scan. The PARSE definition statement must be the first definition statement in the CDF and must precede any CDF records that utilize the special parsing characters that it defines. After the PARSE definition statement is processed, the override parsing characters will go into effect turning the established parsing characters into non-parsing characters. They will simply become data characters.

[*symbol*] PARSE operands **operands**	**CC/ VRM**	**ZOS**	**OS390**	**MVS**
[OPTIONS=(COMCHAR={*comchar*|<u>;</u>},	**N/240**			
CONCHAR={*conchar*|<u>+</u>},	**N/240**			
SEPCHAR={*sepchar*|<u>/</u>},	**N/240**			
FSPCHAR={*fspchar*|<u>~</u>})]	**N/240**			

[*symbol*]

Provides a name for the PARSE definition statement. This parameter is optional and may be any valid assembler language symbol.

[OPTIONS=(COMCHAR=*comchar*|<u>;</u>},
CONCHAR=*conchar*|<u>+</u>},
SEPCHAR=*sepchar*|<u>/</u>},
FSPCHAR=*fspchar*|<u>~</u>})]

The parsing characters that are specified in the OPTIONS= operand must be given as single EBCDIC characters not enclosed in quotes. The following parsing override characters are currently provided:

COMCHAR=*comchar* specifies the <u>comment start character</u> for the CDF. ***comchar*** overrides the semi-colon (;) which is normally used to indicate the start of a comment on a CDF record (after the configuration data). This character can also occur in record column 1 when the entire record is to be treated as a comment, however, an asterisk (*) in record column 1 is normally used for this purpose and cannot be overridden.

CONCHAR=*concHar* specifies the <u>continuation character</u> for the CDF. ***conchar*** overrides the plus-sign (+) which is normally used to indicate that data on the current CDF record is to be continued on the next CDF record.

SEPCHAR=*sepchar* specifies the <u>suboperand separator character</u> for the CDF. ***sepchar*** overrides the forward-slash (/) which is normally used to separate suboperands within an operand entry that accepts multiple suboperands. For example, SVC0=(...,*sluname*/*dteaddr*/*mxtname*,...).

FSPCHAR=*fspchar* specifies the <u>forced space character</u> for the CDF. The blank ( ) is normally used to provide spacing. When CDF records are continued using the plus-sign (+) or the ***conchar*** override character, the CDF parser removes all blanks between the last non-blank character on a record and the continue character. If you want HNAS to leave spaces between the last non-blank character and the continue character on a record, you can specify a forced space character that is used to provide a place holder for blanks that are not to be deleted.

For example, if OPTIONS=FSPCHAR=~ is specified (or allowed to default), you can prevent blanks from being deleted on a continuation record as follows:

```
BUILD TRCTRAP=(ALRMLIST=(NAS8210I/ +
                         LU LA68~~~~~~+
                         ON PU P0AEMERC/ +
                         25)
```

This would be processed as if the following were entered on a single record:

```
BUILD TRCTRAP=(ALRMLIST=(NAS8210I/LU LA68      ON PU P0AEMERC/25)
```

Note that this example illustrates the use of the TRCTRAP= operand to trap message NAS8210I with the given embedded data.

```
0         1         2         3         4         5         6
0123456789012345678901234567890123456789012345678901234567890
NAS8210I ACTLU ENDED FOR LU LA68     ON PU P0AEMERC POWER OFF
```

If a forced space character were not used, the continued CDF records:

```
BUILD TRCTRAP=(ALRMLIST=(NAS8210I/ +
                         LU LA68      +
                         ON PU P0AEMERC/ +
                         25)
```

would not yield the desired result and would be parsed as the single record:

```
BUILD TRCTRAP=(ALRMLIST=(NAS8210I/LU LA68ON PU P0AEMERC/25)
```

BUILD Definition Statement

The BUILD definition statement is used to delimit the start of the X.25 Host NAS configuration and to convey global information to other definition statements in the configuration. The BUILD definition statement is required. If omitted, all other configuration definition statements will abort.

[*symbol*] BUILD operands operands	CC/ VRM	ZOS	OS390	MVS
[ALRMCART=*token*]	230			
[ALRMFLTR=({P[URGE]|S[UPPRESS]|A[LLOW]}, *alrmlist*)]	210 C/220			
[ALRMLMTS=NONE| ({*seconds*|30}, {*ilmt*|20}, {*dlmt*|10}, {*almt*|20}, {*wlmt*|25}, {*elmt*|30}, {*slmt*|30}, {*mlmt*|10})]	220 C/240			
[APPLNAME={*vbldname*|NONE}]	A/220			
[BFRLMT={*count*|8*VCLMT}]				
[BFRSIZ={*size*|196}]	C/240			
[CONCMDQ={*ddname*|(*options*,*cmdlist*)}]	210 C/240			
[CONLMT={*count*|1}]	C/230			
[CONPRMT={'*text*'|'ENTER COMMAND:'}]	114			
[CONPSWD={NONE|*chars*}]				
[EXEC={*ddname*|(*options*,*cmdlist*)}] *** EXEC= is an alias for CONCMDQ= ***	A/240			
[IDLETO={*minutes*|0}]	114			
[LOGTAB=*name*]	210			
[MSGLMT={*count*|2*socketcount+1}]	112			N
[NASNAME={*name*|tcbaddr}]				

[*symbol*] BUILD operands **operands**	**CC/ VRM**	**ZOS**	**OS390**	**MVS**
[OPTIONS=(ALRMSGTXT={SHORT|LONG} LUBLTCNT={*slotcount*|40}, LUDRPOOLCNT={*lucount*|0}, TIMESTAMP={HH:MM:SS[.T[H[M[I]]]]| STD|EXT}[+D], WTOROUTCDE(ALRM)={*alrmcode*|8}, WTOROUTCDE(CONS)={*conscode*|8}, WTOROUTCDE={*bothcodes*|8})]	A/240 220 230 N/240 A/230 A/230 A/230			
[PRTLMT={*count*|65535}]				
[PRTSWLST=({LOOP|STOP},SWITCHAFTERINIT, {SWITCHAT*hh*00| SWITCHAT6AM|SWITCHATMIDDAY| SWITCHAT6PM|SWITCHATMIDNIGHT} {*ddname*1|DYNAMIC=*outclass*},... ...,{*ddname*n|DYNAMIC=*outclass*})]	N/240			
[PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)]	A/240			
[SCHEDULE={*ddname*| (*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)}]	A/240			
[TCPNAME={*name*|TCPIP}]				
[TRCLMT={*count*|4000}]	C/220			
[TRCTRAP= (ALRMLIST=(*id*1/*dat*1/*sod*1,..., *id*n/*dat*n/*sod*n), 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*)})]	A/230 C/240 A/230 N/240			
[USSTAB={*name*|ISTINCDT}]	114			
[VCLMT={*count*|+VCLMT}]				

[*symbol*]

Provides a name for the BUILD definition statement. This parameter is optional and may be any valid assembler language symbol.

[ALRMCART=*token*]

(new for V2R3M0)

Specifies a token that will be used to direct alarm messages. You may specify a ***token*** value from 1 to 8 alphanumeric characters in length. Normally, ***token*** will be the name of a user (address space) that will process the alarm WTOs.

If a ***token*** is provided, it will be passed in the CART= operand of all alarm WTO's. In this way, you can channel alarm messages to a specific system component like NETVIEW. If a ***token*** value is omitted, no alarm WTO's channeling will be provided.

Note: In order for the operating system to process the **WTO CART=*token*** parameter, the SHOWERR or SHOWON option must be in effect.

Note: Normally, a command token is presented to HNAS in the Command Input Buffer (CIB) via a MODIFY *hnas* request. This token is then passed back to the requesting user via the **WTO CART=*token*** parameter so that the command and its response can be correlated. Because alarm messages are unsolicited, there is no command token. The ALRMCART= operand is used to supply the command token for alarm messages.

[ALRMFLTR=({P[URGE] | S[UPPRESS] | A[LLOW]},*alrmlist*)]

(new for V2R1M0)
(changed for V2R4M0)

Specifies a default alarm message disposition and a list of HNAS alarm message identifiers that will be filtered based on the disposition you specify.

The first ALRMFLTR= suboperand is the default disposition that applies to *most* alarm messages generated by HNAS (see table of **FORCED** alarms below for exceptions).

alrmlist defines a list of alarm IDs for which the action has to be different from the default disposition. The ***alrmlist*** identifiers are represented by **NAS*innns*** alert message structures. The ***innns*** fields can be represented by various filtering options for i-component identifier, nnn-message number and s-severity code values. Please refer to the Message Identification Layout section located in the Alert Messages chapter of the HNAS Messages and Codes manual. The default disposition indicates the action that will be taken when an alarm message is issued and the alarm ID is not specified in the ***alrmlist***.

Note: The default disposition also provides the action that is taken when alarm ID is specified in the ***alrmlist*** but it does not have an override disposition.

PURGE, as the first suboperand, indicates that alarm messages not listed in ***alrmlist*** *will not be* displayed on SYSCONS or logged in SYSPRINT. If ***alrmlist*** is omitted, *no* alarm messages will be displayed on SYSCONS or logged in SYSPRINT.

SUPPRESS, as the first suboperand, indicates that alarm messages <u>not listed in ***alrmlist***</u> *will not be* displayed on SYSCONS but will be logged in SYSPRINT. If ***alrmlist*** is omitted, *no* alarm messages will be displayed on SYSCONS but all will be logged in SYSPRINT.

ALLOW, as the first suboperand, indicates that alarm messages <u>not listed in ***alrmlist***</u> *will be* displayed on SYSCONS and logged in SYSPRINT. If ***alrmlist*** is omitted, *all* alarm messages will be displayed on SYSCONS and logged in SYSPRINT.

Note: If ALLOW is in effect, many alarm messages may still not be displayed on SYSCONS depending on the SHOW action in effect. For example, SHOWERR will allow only error alarm messages (severity of W|E|A|S) to be routed to SYSCONS while informational alarm messages (severity of I|D) will not be routed to SYSCONS.

The ***alrmlist*** you specify must contain valid HNAS alarm message identifiers each of which must be separated by a comma. The ***alrmlist*** has the following form:

***alrmlist=id*i({P|S|A|FC|FU}),...,*id*n({P|S|A|FC|FU})**

where the ***id*i** values are 8 character HNAS alarm identifiers **NAS*innns*** (e.g., NAS2031W) and **({P|S|A|FC|FU})** is an override disposition that supersedes the default disposition given by the first **ALRMFLTR=** suboperand. If you omit the **({P|S|A|FC|FU})** override disposition for an ***id*i** value, the default disposition wins. The ***id*i** values may contain ***** wildcard characters in order to filter a collection of alarms.

Note: The (FC) and (FU) message ID suffixes allow various alarm messages to be forced conditionally to SYSCONS (subject to SHOW state) or forced unconditionally to SYSCONS (SHOW state ignored). For example, if ALRMFLTR=(...,NAS2020I(FC),NAS0299I(FU),...) is specified, NAS2020I alarm messages will be sent to SYSCONS except when SHOWOFF is effect while NAS0299I alarm messages will be sent to SYSCONS regardless of the SHOW state.

Note: Monitor messages **NAS*innn*M** (produced by **MON TAP**) can be purged (not written to SYSPRINT) via the PURGE option although the SUPRESS option is ignored because Monitor messages are not written to SYSCONS.

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

You may specify from 1 to 128 ***id*i** values.

Note: The ALRMFLTR= operand limit was changed from 16 to 128 starting with V2R4M0.

If the **ALRMFLTR=** operand is not specified, a default action of **ALLOW** will be used.

Important Note: HNAS alarm messages are always written to the SYSPRINT log file regardless of the **ALRMFLTR=** operand values in affect unless **PURGE** is specified.

`Example1:` `ALRMFLTR=(ALLOW,`**`NAS2***I(P)`**`)`

Specifies that HNAS will display all alarms except TCPIP informational alarm messages, that is, messages of the form **NAS2xxxI**. TCPIP informational alarm messages are useful when debugging TCPIP activity but can usually be ignored during normal HNAS operation. We recommend filtering these alarms out from SYSCONS and SYSPRINT as example1 shows. Implementation of the PURGE option for these alert message types can dramatically reduce the SYSPRINT file size.

Example2: `ALRMFLTR=(SUPPRESS,NAS2****(A))`

Specifies that HNAS will display only TCPIP alarm messages, that is, messages that start with **NAS2**. This type of filtering could be used exclusively to debug a TCPIP only condition where other message types weren't required during the debug process.

We suggest that users initially set the s-value for the alarm filters to an asterisk (*) so that all severity levels are either **A**llowed or **S**uppressed or **P**urged (i.e. NASnnnn*(S), NASnnnn*(A) or NASnnnn*(P)). This will accommodate the occasional reassignment of Severity Levels as deemed appropriate in HNAS maintenance releases or upgrades. All alert message severity level reassignments will be documented in APAR memo's or located in the appropriate VnRnMn Migration section.

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

For example, coding ALRMFLTR=(PURGE,NAS3799I) is the same as coding ALRM-FLTR=(PURGE,NAS3799I(P)).

To purge all but the NAS3799I message, code ALRMFLTR=(PURGE,NAS3799I(A)) or ALRMFLTR=(PURGE,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* ALRM-FLTR operand (default disposition) was PURGE. We recommend using ALLOW as the default disposition and specifying (P) or (S) as appropriate for the ***id*i** values in the list. If PURGE 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 affect, 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 ALRMFLTR=(P,NAS*(A)). Another way to do this is to specify **SHOW ON** (start parameter **SHOWON**) and ALRMFLTR=(A,NAS****I(P)). SHOWON makes informational alarms subject to full alarm filtering not just the default disposition.

Note: Alarm filter values may be added, deleted, modified and displayed using the **ALARM FILTER=** console command.

Note: Some **FORCED** alarm messages cannot be filtered by the ALRMFLTR= operand. These are alarms that are considered critical, and as such, must be written to the SYSCONS and logged in SYSPRINT. The following table lists those alarms that cannot be filtered. If specified in the ***alrmlist***, they will be skipped during alarm filter processing.

Message ID	Message Description
`NAS0001I`	HNAS initialization complete alert message
`NAS003xI`	HNAS shutdown messages
`NAS0050A`	Tracing suspended or snapshot dump taken by *consname*
`NAS0060W`	Tracing resumed or snapshot dump resumed by *consname*
`NAS0070W`	Trace trap action status changed by *consname*
`NAS0071W`	Trace trap action status
`NAS0120I`	SYSCONS error and info alarms enabled by *consname*
`NAS0121W`	SYSCONS non-forced alarms disabled by *consname*
`NAS0122I`	SYSCONS error alarms enabled by *consname*
`NAS0130W`	SYSCONS alarm options modified by *consname*
`NAS0201E`	SYSPRINT log record limit reached, logging terminated
`NAS0204E`	SYSPRINT ABEND, logging terminated
`NAS0205I`	SYSPRINT logging terminated by SHUTDOWN
`NAS0207W`	SYSPRINT PRTSWLST= log datasets all used
`NAS0208I`	SYSPRINT PRTSWLST= log dataset opened
`NAS0209E`	SYSPRINT PRTSWLST= log dataset could not be opened
`NAS0210I`	SYSPRINT PRTSWLST= log switch requested
`NAS0210I`	SYSPRINT logging enabled by *consname*
`NAS0210W`	SYSPRINT trace logging enabled by *consname*
`NAS0211I`	SYSPRINT trace logging disabled by *consname*
`NAS0211W`	SYSPRINT logging disabled by *consname*
`NAS0230W`	SYSPRINT logging options modified by *consname*
`NAS0910I`	End of day crossover message issued at midnight
`NAS920xI`	HNAS authorization messages

```
[ALRMLMTS=NONE|
          ({seconds|30},
          {ilmt|20},
          {dlmt|10},
          {almt|20},
          {wlmt|25},
          {elmt|30},
          {slmt|30},
          {mlmt|10})]
                                                   (new for V2R2M0)
                                              (apar/new for V2R2M0)
```

Specifies a time interval during which HNAS alarm messages are counted and a list of alarm limits for the counted alarms as a function of the alarm severity. HNAS alarm messages that pass ALRMFLTR= operand filter processing may now be limited from display at the system console to prevent depletion of the system WTO buffer pool. When an alarm message count reaches its limit, the alarm message is no longer displayed at the system console.

NONE specifies that alarm messages are not to be counted and an alarm logging table is not to be created. All messages will be displayed on the system console if they pass ALRM-FLTR= operand filter processing.

seconds, as the first suboperand, indicates a time interval in seconds during which alarms are counted. If a counted alarm reaches its limit during the time interval, it is no longer displayed on the system console for the duration of the time interval. The alarm counts are reset at the start of the next time interval. You may specify a ***seconds*** value between 15 and 254. If ***seconds*** is not specified or if the specified value is invalid, a default value of **30** will be used.

ilmt represents the limit for **I**nformation messages. You may specify a ***ilmt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***ilmt*** is not specified or if the specified value is invalid, a default value of **20** will be used.

dlmt represents the limit for **D**efault processing messages. You may specify a ***dlmt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***dlmt*** is not specified or if the specified value is invalid, a default value of **10** will be used.

almt represents the limit for user **A**ction required messages. You may specify an ***almt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***almt*** is not specified or if the specified value is invalid, a default value of **20** will be used.

wlmt represents the limit for **W**arning messages. You may specify a ***wlmt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***wlmt*** is not specified or if the specified value is invalid, a default value of **25** will be used.

elmt represents the limit for **E**rror messages. You may specify an ***elmt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***elmt*** is not specified or if the specified value is invalid, a default value of **30** will be used.

slmt represents the limit for **S**evere error messages. You may specify an ***slmt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***slmt*** is not specified or if the specified value is invalid, a default value of **30** will be used.

mlmt represents the limit for **M**iscellaneous messages. You may specify an ***mlmt*** value between 0 and 254. If 0 is specified, it is treated as no limit. If ***mlmt*** is not specified or if the specified value is invalid, a default value of **10** will be used.

For example, specifying **ALRMLMTS=(60,75,20,40,50,60,60,20)** will set 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.

Note: Alarm limit values may be added, deleted, modified and displayed using the **ALARM LIMITS=** console command. The alarm logging table may be cleared (all entries removed), reset (all counters reset for current entries) and displayed using the **ALARM LOG=** console command.

Note: The **ninth (9th)** ALRMLMTS= suboperand (the alarm logging table limit value) has been retired effective with APAR 2400089. This was done because the alarm logging table was allocated after the CDF scan completed which caused a number of runtime alarms (like NAS92*xxs* AUTH alarms) to not be logged. HNAS has been modified to allocate the alarm logging table immediately after it starts with a default entry limit of 512. The limit, which was previously specified in the CDF as the ninth (last) suboperand of the ALRMLMTS= operand, is no longer accepted. If a value is specified, the following configuration alert message is generated:

```
NAS1101D BUILD ALRMLMTS LOGGING TABLE LIMIT PARAMETER
NAS1101D HAS BEEN RETIRED, 0512 ALWAYS USED
```

Prior to APAR 2400089, a user could specify a value of zero (0) for the alarm logging table limit which would prevent the table from being created. Since the alarm logging table is an important component for debugging problems, this change means that the table will ALWAYS be generated.

[APPLNAME={*vbldname*|NONE}]

(added for V2R2M0)

Specifies a name that will be used for the VBUILD statement in the AMNF that is produced by the HNAS FASTRUN process.

The ***vbldname*** value you specify can be any valid assembler language symbol.

If a ***vbldname*** value is not specified or if the specified value is invalid, the VBUILD statement name will come from the NASNAME= operand as it did in the past. If the NASNAME= operand is also omitted, the VBUILD statement will be produced without a name.

If **NONE** is specified, the VBUILD statement will be produced without a name.

[BFRLMT={*count*|8*VCLMT}]

Specifies the number of I/O buffers to be reserved for HNAS operation. HNAS uses the BFRLMT= and BFRSIZ= operands to compute the amount of mainframe memory that will be allocated for the system buffer pool.

You may specify a ***count*** value between 50 and 65535. If a ***count*** value is not specified or if the specified value is invalid, a default value of **8*VCLMT** will be used. The default value is linked to the total number of virtual circuits that HNAS can support but may be more than is actually required.

[BFRSIZ={*size*|196}]

Specifies the size (in bytes) of each buffer in the HNAS buffer pool.

You may specify a ***size*** value between 196 and 9362. If a ***size*** value is not specified or if the specified value is invalid, a default value of **196** will be used. The default value was optimized for environments running with 128 byte data packets (128+68=196).

XOT HNAS buffers are packet oriented so a single buffer will never contain more than one packet although a packet can span multiple buffers. Each header buffer requires 68 bytes for non-data control information. If you know that all of your X.25 sessions will be running with a packet size greater than 128 and if packets are generally full or chained you can further improve efficiency by coding a larger buffer size to accommodate the larger packets.

Note: Prior to V2R4, the packet header size was 52 bytes so that a BFRSIZ=180 would accommodate a full packet of 128 bytes in length. Starting with V2R4, the packet header size has been increased by 16 bytes to 68 bytes for diagnostic purposes. For this reason, you must add 16 bytes to the value you previously specified for BFRSIZ=. If HNAS detects that the new packet header length has not been accounted for, it will automatically add 16 bytes to the value specified for BFRSIZ= and the following message will be issued:

```
NAS1121D BUILD BFRSIZ=dddd AUTOMATICALLY INCREASED TO dddd,
         DEBUG FIELD ADDED
```

The BFRSIZ= value will not be increased by 16-bytes and this message will be withheld if the value you code for BFRSIZ= already accounts for the new 68 byte packet header, that is, it is one of the following values:

196 = 68 + 128
324 = 68 + 256
580 = 68 + 512
1092 = 68 + 1024
2116 = 68 + 2048
4164 = 68 + 4096
9362 = TCPIP transfer limit

Note that the X.25 packet size is always given as a power of 2 via facility code 42 therefore intervening values do not have to be tested.

Note: The automatic adjustment of the BFRSIZE= value and accompanying NAS1121D message was introduced into 240 via APAR 2400002.

[CONCMDQ={*ddname*|(*options*,*cmdlist*)}]

(new for V2R1M0)
(changed for V2R4M0)

Specifies a list of Console Subsystem commands that are executed when HNAS starts. You may specify either an *inline* list of commands or a *ddname* that identifies a file containing a list of commands.

You may specify an *inline* list of commands in a sublist as follows:

(***options***,***cmdlist***)

The ***cmdlist*** (*cmd*1,...,*cmd*n) you specify must be valid Console Subsystem commands each of which must be separated by a comma and enclosed in quotes if the command contains blank characters.

Example: `CONCMDQ=(DPARM,DMAP,'TRCMCH ICR','EXEC` *ddname*`)`

This example specifies that HNAS will execute the DPARM, DMAP and TRCMCH ICR console commands and execute the command list identified by *ddname* automatically before any operator entered commands.

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: HNAS will force **DNAS** to be executed at startup so there is no requirement to include it in the ***cmdlist***.

Note: CONCMDQ= can include the **EXEC *ddname*** command. If **EXEC** is specified in the ***cmdlist***, it should be the last command in the list. This is because HNAS will stop executing the ***cmdlist*** and start executing the **EXEC *ddname*** command when it detects the EXEC command (the same memory is used to remember the CONCMDQ= and the EXEC command list values). The CONCMDQ= command list is considered to be the initial command list. Please refer to the **EXEC *ddname*** console command for additional information.

> Plans are currently underway to eliminate the ‘same memory used’ condition which will allow multiple **EXEC *ddname*** to be provided under CONCMDQ= or embedded in a command lists to eliminate this restriction. This will be provided as an Enhancement APAR.

As an alternative to specifying an *inline* command list, you may specify a *ddname* that identifies a file that contains a list of commands. A *ddname* command list can contain the following records:

Comments are allowed and start with an asterisk (*), semi-colon (;) or pound-sign (#) in record column 1. Comments can also appear on a command line but must start with a semi-colon after the command. Comments that start with # are displayed in SYSPRINT and on SYSCONS based on the current PRNT and SHOW states, respectively.

A single command can be specified on a single data record as follows:

cmd ; *comment*

Multiple commands can be specified on a single data record as follows:

(*cmd*1, ... , *cmd*n) ; *comment*

The following is a sample command list that can be provided by **CONCMDQ=*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
  NOPURGEONERROR    ; NO PURGE ON ERROR OPTION
  TRCALL STOP      ; 1ST QUEUED COMMAND TO STOP ALL ACTIVE TRACES
  TRCPCE ALLON     ; 2ND QUEUED COMMAND TO START TCPIP TRACING
  TRCPRNT          ; 3RD QUEUED COMMAND TO START TRACE PRINT LOGGING
  PING DMY001      ; 4TH QUEUED COMMAND TO PING REMOTE DMY001
  PAUSE 60         ; 5TH QUEUED COMMAND TO PAUSE FOR 1-MINUTE
  TRCPRNT OFF      ; 6TH QUEUED COMMAND TO STOP TRACE PRINT LOGGING
  TRCPCE ALLOFF    ; 7TH QUEUED COMMAND TO STOP TCPIP TRACING
```

The SYSPRINT log now contains an HNAS TCPIP trace of the PING exchanges.

Note: The CONCMDQ=*ddname* support was introduced into 240 via APAR 2400080.

Note: Specifying CONCMDQ=(EXEC=*ddname*) instead of CONCMDQ=*ddname* means that the given *ddname* will not be checked for validity by the configuration process because the commands you code in the CONCMDQ= operand list are not checked until the individual commands are executed. Conversely, when CONCMDQ=*ddname* is specified, the configuration process validates the existence of *ddname*.

The maximum number of commands depends on the available queued storage space. A total of 512 bytes are reserved for queued console commands. Each blank is counted as a text character. The 512 byte maximum also includes a length byte for each queued command. For the example above, 22 bytes are required to queue the commands: 6 for DPARM (length byte = 5), 5 for DMAP (length byte = 4) and 11 for TRCMCH ICR (length byte = 10).

Note: The CONCMDQ= operand queue area is also used to save commands provided by the EXEC and QE (QUITEXEC) console commands.

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=.

[CONLMT={*count*|1}]

(changed for V2R3M0)

Specifies the number of simultaneous remote console sessions that HNAS can support. Remote consoles connect to HNAS as ASCII TTY-like devices over an ITI/XTP or ITI/XOT session. However, rather than accessing a host application via the PCNE function, remote console sessions communicate with the HNAS console subsystem. Please refer to Console book section for more information on console access.

You may specify a ***count*** value between 0 and 11. If a ***count*** value is not specified or if the specified value is invalid, a default value of **1** will be used. If 0 is specified, remote console access will be barred and the omitted console password message (NAS1101W) will be suppressed.

Note: Remote console connections will be rejected (call will be cleared) if you do not specify the **RMTCONS|RMTCONP** start parameter or issue the **RMTCONS ON|PRIV** console command.

[CONPRMT={'*text*'|'ENTER COMMAND:'}]

(new for V1R1M4)

Specifies a Console Subsystem input prompting message.

You may specify a ***text*** value as a string of up to 16 characters. The text string must be enclosed in quotes if it contains blanks. Each blank is counted as a text character.

If a ***text*** value is not specified or if the specified value is invalid (text string is too long), a default prompting message of **ENTER COMMAND:** will be used.

[CONPSWD={NONE|*chars*}]

Specifies an access password for remote console sessions.

If a ***chars*** value is specified as NONE, no password protection will be provided for remote console access.

You may specify a ***chars*** value of up to 8 non-blank characters. If a ***chars*** value is not specified or if the specified value is invalid (text string is too long), remote console access will be denied.

Note: Remote console connections are limited to non-privileged (display only) status if you do not specify the **RMTCONP** start parameter or issue the **RMTCONS PRIV** console command. This also applies if **CONPSWD=NONE** is specified.

If privileged remote console access is allowed, the console operator must select this mode of operation by entering the console password backwards. If the backwards and forwards versions of the console password are the same (e.g., CONPSWD=WOW), the remote console session will operate in non-privileged mode only.

[EXEC={*ddname*|(*cmdlist*)}]

(added for V2R4M0)

Specifies a list of Console Subsystem commands that are executed when HNAS starts. The EXEC= operand is an alias for the CONCMDQ= operand. It is processed exactly the same way. EXEC= was added to provide name symmetry between the configuration operand and the EXEC console command similar to what was done for the SCHEDULE= operand and the SCHEDULE console command. Please see the description the CONCMDQ= for more information.

Note: The EXEC= operand support was introduced into 240 via APAR 2400080.

[IDLETO={*minutes*|<u>0</u>}]

(new for V1R1M4)

Specifies an inactivity timeout value for all HNAS virtual circuit connections (PVC and remote console sessions are excepted). If a VC does not receive or send data in the specified ***minutes*** interval then the virtual circuit call is cleared with cause=198. A zero value indicates that inactivity time-outs are not provided. The value coded on BUILD may be overridden by IDLETO= parameters coded on MCH, MXT or SPU REMOTE statements. Zero is assumed if the parameter is omitted or if the value coded is not in the range 0 to 255.

[LOGTAB=*name*]

(new for V2R1M0)

Specifies the name of a LOGON table that will be used as the system default LOGON table for PCNE and/or PAD application LOGON interpret processing by the MCHSOL routine. For more information on MCHSOL services, please see the description of the APPLNAME= operand of the REMOTE definition statement on page 4-74 of this document.

Note: If both **LOGTAB=** and **USSTAB=** are specified, **LOGTAB=** is processed first by MCHSOL.

The ***name*** value you specify can be any valid assembler language symbol.

If a ***name*** value is not specified or if the specified value is invalid or if the specified value is not found in the library identified by the **VTAMLIB DD** statement, only the **USSTAB=** operand will be used for LOGON interpret processing.

[MSGLMT={*count*|2*socketcount+1}]

(new for V1R1M2)

Specifies the maximum number of TCP/IP message requests that HNAS can issue before it must wait for a reply. For each TCP socket, there can be at most three (3) TCP/IP requests outstanding: control, send and receive. The number of TCP sockets (**socketcount**) that HNAS will service is based on the number of LOCAL and REMOTE definition statements in the CDF. This number can be computed using the following table.

Statement	TYPE	Number of TCP Sockets
LOCAL	XTP	SOCLMT/2000+1
LOCAL	XOT	SOCLMT/2000+1
REMOTE	XTP	1 (with same IPADDR and PORT)
REMOTE	XOT	VCLMT

The TCP/IP message limit is used to govern the number of simultaneous requests that HNAS can issue to the TCP/IP stack. It is used across all router connections so that HNAS does not overrun the TCP/IP stack.

You may specify a ***count*** value between 3 and 16383. If a ***count*** value is not specified or if the specified value is invalid, a default value of **2*socketcount+1** will be used. The default value is linked to the total number of TCP sockets that HNAS will support.

Note: We recommend that you allow the MSGLMT= operand to default to the HNAS calculated count. If a count is set too low the Network VC or VTAM LU activity may enter pseudo slowdown mode which can prevent some processes from completing in a timely manner.

[NASNAME={*name*|tcbaddr}]

Specifies the name of the HNAS address space as it appears to the TCP/IP address space. This name is used when HNAS establishes its presence to the TCP/IP program product. It can also appear in the console subsystem's input prompt (refer to **PFXWTO** start parameter or **PFXWTO ON** console command for more information).

You may specify a ***name*** value of up to 8 non-blank characters. If a ***name*** value is not specified or if the specified value is invalid (text string is too long), a default name will be generated from the Task Control Block (TCB) address. This guarantees uniqueness.

```
[OPTIONS=(ALRMSGTXT={SHORT|LONG}
          LUBLTCNT={slotcount|40},
          LUDRPOOLCNT={lucount|0},
          TIMESTAMP={HH:MM:SS[.T[H[M[I]]]]|STD|EXT}[+D],
          WTOROUTCDE(ALRM)={alrmcode|8},
          WTOROUTCDE(CONS)={conscode|8},
          WTOROUTCDE={bothcodes|8})]
                                                    (new for V2R2M0)
                                                (changed for V2R3M0)
                                                (changed for V2R4M0)
```

Specifies special processing options for this BUILD definition statement that are global to the entire configuration.

ALRMSGTXT={SHORT|LONG} allows you to select the long (default) or short format for selected alarm messages. The 'SHORT' versions of these messages consolidates information on a single line. The new OPTIONS=ALRMSGTXT= operand effects various error, informational and traces messages.

For XOT (**NAS7*nnns***), the following callin/callout alert and trace messages are affected by this option:

NAS7716W, **NAS7717W**
Each of the above messages requires 2 messages when LONG is coded and 1 message when SHORT is coded. See specific Alert message for the layouts.

NAS7718T, **NAS7719T**, **NAS7798T**
These messages, which are sent only to SYSPRINT, are generated when the TRCMCH ICR/OCR ON command is issued. Three NAS7798T message follow each NAS7718T (ICR) or NAS7719T (OCR) message when LONG is coded. The messages provide detailed information about call request packets received and sent by HNAS. When SHORT is in effect, a single **NAS7730I** (ICR) or **NAS7731I** (OCR) message is sent to SYSPRINT and the operator's console. The intent of this is to allow programs that monitor the job log to process call request packet information.

For TCPIP (**NAS2*nnns***), VC (**NAS5*nnns***) and QLLC (**NAS8*nnns***) alarm messages that have the following LONG format:

```
NAScnnns comp=iii.iii.iii.iii(port) SOCKID=sockid PCEID=pceid NAME=compname
NAScnnns information text
```

The new SHORT format will appear as follows:

```
NAScnnns comp=iii.iii.iii.iii(port) NAME=compname information text
```

As you can see, the SOCKID= and PCEID= fields have been removed for the SHORT format so the alarm message can fit on a single line.

For these messages, ***c*** is the component ID (2|5|8), ***nnn*** is the message number, ***s*** is the message severity (I|W|E|S) and ***comp*** is the component type (SERVER or CLIENT).

Example:

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

becomes (when ALRMSGTXT=SHORT is in effect):

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

Note: OPTIONS=ALRMSGTXT={SHORT|LONG} support was introduced into 240 as Enhancement APAR 2400068.

LUBLTCNT={*slotcount*|40} specifies the number of **slots** that are to be reserved in the SLU buffer list table for VTAM data transfers. Each slot accommodates a data address and data count which allows transfers of non-contiguous data. In releases prior to V2R2M0, the number of slots was maintained as an internal value (40 for V2R1 and 30 for V1). Because some transactions can utilize buffer chains which are longer than the internal slot count, the slot count had to be parameterized.

When the ***slotcount*** value is too small, inbound GATE file transfers can fail when the SLU buffer list table is not large enough to hold all the packets in an M-bit chain. In this case, the PLU will abort the incomplete transfer. An application like CFT, for example, will issue an ‘Invalid FPDU’ message when this condition occurs while HNAS will issue a NAS3720S alert message at the same time.

This condition normally manifests itself when HNAS receives an extremely large message sequence that occupies more M-bit chained packets than the ***slotcount*** value. This condition can also happen when excessive TCP/IP buffer fragmentation occurs that causes HNAS to populate the SLU buffer list table with more buffers than the list can accommodate.

You may specify a ***slotcount*** value between 25 and 500. If the **LUBLTCNT=** option is not specified or if the specified value is invalid, a default value of **40** (slots) will be used.

LUDRPOOLCNT={*lucount*|0} specifies the number of LUs that are to be reserved in the LU Dynamic Reconfiguration Pool (LUDRPOOL) for allocation by the MRMT console command.

You may specify a ***lucount*** value between 0 and 65534. If the **LUDRPOOLCNT=** option is not specified or if the specified value is invalid, a default value of **0** will be used. In this case, you will be unable to add entries to any SPU LUNAME= operand list or any MCH SVC*i*= operand list.

For TYPE=SPU REMOTE definition statements, you may enter the following console command to add, delete or modify entries in the LUNAME= operand list:

```
MRMT LUNAME=(locaddr,{slunm/rpace/space/applid/mxtname|*},...,
                          :
                          {slunm/rpace/space/applid/mxtname|*})
```

Where: ***locaddr*** is the starting LOCADDR entry in the LUNAME= operand.
slunm is an existing or new SLU name for the selected entry.
rpace is a changed or new receive pacing value for the selected entry.
space is a changed or new send pacing value 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 entry is to be removed from LUNAME= operand.

If ***locaddr*** is greater than the existing LUNAME= operand list size, LUs will be allocated from the LUDRPOOL 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 ***slunm*** 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 (UNBIND, CLOSE ACB) if its ***slunm*** is being changed or deleted.

For TYPE=XTP|MCH REMOTE definition statements, you may enter the following console command to add, delete or modify entries in the SVC*i*= operand list (*i*=0,4,5):

```
MRMT SVCi=slunm={newnm|*}/optional parms...
```

Where: ***slunm*** is an existing or new SLU name for the selected entry.
newnm is a new name for an existing SLU or is a new SLU. *slunm* and *newnm* must be the same value for the selected entry (*slunm=slunm* must be specified).
***** indicates that the selected entry is to be removed from SVC*i*= operand.

If *slunm=slunm* does not exist, an LU will be allocated from the LUDRPOOL to satisfy the addition of the new list entry. Each SVC*i*= operand can support a maximum of 511 LUs. If an SVC*i*= 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 ***slunm*** represents a new name for an existing or new SVC*i*= 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 selected SVC*i*= entry, it will be deactivated (UNBIND, CLOSE ACB) if its ***slunm*** is being changed or deleted.

TIMESTAMP=HH:MM:SS[.T[H[M[I]]]] specifies that the HNAS SYSPRINT log timestamp 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 PRNTDATE ON).

Note: The format established by the TIMESTAMP= suboperand will not take effect until after HNAS initialization completes and the NAS0001I INITIALZATION COMPLETE message is issued.

WTOROUTCDE(ALRM)={*alrmcode*|8} specifies the routing code that will be used for HNAS alarm message WTOs.

You may specify a ***alrmcode*** value between 0 and 128. If the **WTOROUTCDE(ALRM)=** option is not specified or if the specified value is invalid, a default value of **8** (teleprocessing control) will be used. In this case, you will not be able to prevent HNAS alarm messages from going to SYSLOG if SYSLOG is defined as a HARDCOPY console. This is because WTO routing codes 1,2,3,4,7,8,10 and 42 cannot be dropped from the HARDCOPY ROUTCODE= list.

WTOROUTCDE(ALRM)={*conscode*|8} specifies the routing code that will be used for HNAS console command output WTOs.

You may specify a ***conscode*** value between 0 and 128. If the **WTOROUTCDE(CONS)=** option is not specified or if the specified value is invalid, a default value of **8** (teleprocessing control) will be used.

Note: When SHOWCONS is in effect, console output will always be routed to the HARD-COPY device because the output is marked as a response. The only way to prevent this and allow a response to be filtered by routing code is to specify CMDLEVEL(NOCMDS) for the HARDCOPY device. However, it may not be possible to use the NOCMDS option in all environments. For more information, please refer to the discussion of the CONSOLxx member in the MVS Initialization and Tuning Reference (SC28-1752).

WTOROUTCDE={*bothcodes*|8} specifies the routing code that will be used for both HNAS alarm message WTOs and console command output WTOs.

You may specify a ***bothcodes*** value between 0 and 128. If the **WTOROUTCDE=** option is not specified or if the specified value is invalid, a default value of **8** (teleprocessing control) will be used for alarm message WTOs and console output WTOs.

[PRTLMT={*count*|65535}]

Specifies the number of records that can be written to the HNAS SYSPRINT log file. When this limit is reached, all subsequent SYSPRINT log requests are ignored.

This output record limit is provided to prevent HNAS from consuming excessive system spool resources when HNAS is 1) left running for extended periods 2) excessive session connect activity without alarm limits set or 3) when diagnostic alarm, event or trace activity is generating more SYSPRINT records than usual and to circumvent an HNAS system ABEND code B37 that can occur when an attempt is made to write into a dataset that is full. If you set-up the HNASXEQ JOB to write to a dataset file instead of the system spool please ensure that you set the PRTLMT= operand on the BUILD definition statement to a record count that will not exceed the maximum size of the SYSPRINT dataset.

HNAS will write a message to the local console each time 16384 SYSPRINT records are written (before the limit is reached) or if 262144 SYSPRINT records are lost (after the limit is reached). This console message can be suppressed using the **SHOWOFF** start parameter or **SHOW OFF** console command.

The SYSPRINT dataset can be the JES spool, an OS/390 or MVS sequential dataset. The PRTLMT counter is reset each time a new log file is opened (Multiple SYSOUT support).

You may specify a ***count*** value between 4095 and 16777215. If a ***count*** value is not specified or if the specified value is invalid, a default value of **65535** will be used.

Note: The PRTLMT= operand maximum value was changed from 134217727 to 16777215 for PRTSWLST=DYNAMIC support via 240 as Enhancement APAR 2400001.

Note: When a JES spool file is used as the SYSPRINT log (SYSOUT=) and the file is closed and then re-opened, it can have a record count that exceeds the PRTLMT= value. This is because DISP=MOD is forced for JES spool files. SYSPRINT datasets can be closed and opened manually using the PRNT CLSOPN *ddname* console command or automatically via PRTSWLST= switching (see below) based on a time value or when the PRTLMT= value is reached. In the latter case, if a previously used SYSPRINT dataset is re-opened, it will contain more records than the PRTLMT= value because existing records will be preserved and additional records will be added to the end of the dataset.

```
[PRTSWLST={LOOP|STOP},SWITCHAFTERINIT,
          {SWITCHAThh00|
           SWITCHAT6AM|SWITCHATMIDDAY|
           SWITCHAT6PM|SWITCHATMIDNIGHT}
          {ddname1|DYNAMIC=outclass},...
          ...,{ddnamen|DYNAMIC=outclass})]
```

(new for V2R4M0)

Specifies a list of static (*ddname*i) or dynamic (DYNAMIC=*outclass*) 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 HNAS will terminate after the CDF scan is complete.

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=(LOOP,SWITCHAFTERINIT,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:

```
NAS0210s DDNAME(S) OF THE FORM DYN#xxxx WERE SPECIFIED, PRTSWLST=DYNAMIC
NAS0210s WILL NOT BE ALLOWED, DDNAME(S) CONFLICT
```

This message can be issued with ***s***=W (warning) when HNAS starts. This message can also be issued with ***s***=E (error) 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 that in the latter case where ***s***=E, 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, SWITCHAFTERINIT, SWITCH*hh*00, SWITCHAT6AM, SWITCHATMIDDAY, SWITCHAT6PM and SWITCHATMIDNIGHT 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 (PRTLMT= value is reached), 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.

<u>STOP</u> (the default) specifies that SYSPRINT logging stops when the last DDNAME (*ddname*n) in the PRTSWLST= operand list becomes full.

SWITCHAFTERINIT specifies that logging in the default SYSPRINT dataset is to stop and logging in the *ddname*1 dataset is to start after the **NAS0001I INITIALIZATION COMPLETE** message is issued. This option is provided so that customers can send us an abbreviated HNAS log file if we need to look at the CDF scan and DNAS console command output.

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 SWITCHATMIDNIGHT 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: `PRTSWLST=(LOOP,SWITCHATMIDNIGHT,SYSPRNT1,SYSPRNT2,SYSPRNT3).`

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.

Alarm messages **NAS0207W**, **NAS0208I, NAS0209E** and **NAS0210I** 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.

Note: The DCB parameters for datasets defined in the PRTSWLST= operand are forced to RECFM=VA, LRECL=137 and BLKSIZE=4114.

[PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)]

(added for V2R4M0)

Specifies a start time, stop time and frequency value in seconds that are used to control HNAS heartbeat alarm message generation.

```
        |<---------------------------- start time
        |
        |        |<------------------- stop time
        |        |
        |        |        |<---------- frequency
        |        |        |
        V        V        V
PULSE=(hh:mm:ss,hh:mm:ss,seconds)
```

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 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.

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

[SCHEDULE={*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)}]
(added for V2R4M0)

Specifies either an *inline* list of times and commands or a *ddname* that identifies a file containing a list of times and commands. As an aid to problem diagnosis, sometimes it is necessary to start/stop traces as well as issue display commands at specific times. Prior to the new SCHEDULE operand, this action could only be done manually via operator intervention. The new SCHEDULE= operand was added to allow commands to be scheduled for automatic execution at defined times.

You may specify an *inline* list of times and commands in a sublist as follows:

(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)

As an alternative to specifying an *inline* schedule list, you may specify a *ddname* that identifies a file that contains a list of times and commands. 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). Times can also include asterisks (*) as wildcards so that the same command can be executed at various times during the day (see Note 6 below). A *ddname* schedule list can contain the following records:

Comments are allowed and start with an asterisk (*) or semi-colon (;) in record column 1. Comments can also appear on a command line but must start with a semi-colon after the command.

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 be specified on a single data record as follows:

(*hh*:*mm*:*ss* , *cmd* , ... , *hh*:*mm*:*ss* , *cmd*) ; *comment*

The following is a sample schedule list that can be provided via **SCHEDULE=*ddname***:

```
* THIS IS A SAMPLE SCHEDULED 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 STARTING WITH
* THE FIRST NON-BLANK CHARACTER TO THE LAST NON-BLANK CHARACTER ON
* EACH LINE AFTER THE TIME WHICH FOLLOWS THE OPENING PARENTHESIS '('.
* THE SEARCH FOR A TIME AND COMMAND IS EXECUTED FOR NON-COMMENT
* RECORDS ONLY (*|; IS NOT IN CC1).
*
*           |<-- FIRST NON-BLANK
*           |
```

```
*           |         |<-------- LAST NON-BLANK
*           |         |   |
*           V         V   V
(00:00:00 , ALARM LOG=?)     ; 1ST QUEUED COMMAND TO DISPLAY ALARM LOG AT MIDNIGHT
(12:00:00 , DLU STATE=BOUND) ; 2ND QUEUED COMMAND TO DISPLAY BOUND LUS AT NOON
```

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:

```
NAS1154E SCHEDULE XXXXXXXX COMMAND QUEUE LIMIT count/limit,
                  EXCEEDED, REQUIRED
```

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.

General Notes:

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.

4. The CONCMDQ queue within the console PCE, which is normally used to execute an EXEC *cmdlist*, is also used to execute scheduled commands. The *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 <u>except the low order seconds digit</u>. 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.

   <u>Examples</u>:

   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.

   <u>Examples</u>:

   If you want to run a different schedule list each day, you could structure them as follows (assume HNAS is started on Monday):

```
BUILD 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 HNAS is started during the day):

```
BUILD 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. When the SCHEDULE= operand is decoded during HNAS activation, scheduled commands are suspended until after the **NAS0001I INITIALIZATION COMPLETE** message is issued and after the first CONCMDQ=*cmdlist* processing.

Note: The new SCHEDULE= operand support was introduced into 240 as Enhancement APAR 2400064.

[TCPNAME={*name*|<u>TCPIP</u>}]

Specifies the name of the TCP/IP address space as it appears to the HNAS address space. This name is used when HNAS establishes its server(s) presence to the TCP/IP program product.

You may specify a ***name*** value of up to 8 non-blank characters. If a ***name*** value is not specified or if the specified value is invalid (text string is too long), a default name of **TCPIP** will be used.

Note: Some environments **require** that the HNAS (230 and above) or NASMAIN (220 and below) program be registered under their host resident security subsystem like RACF or Top Secret. This may be required so that the HNAS product can communicate with the host stack TCPIP addresses and processes that are defined in the HNAS Configuration Definition File. Failure to register HNAS with the resident security subsystem in some environments can prevent HNAS from correctly communicating with the TCPIP stack. We suggest that you discuss this potential requirement with your system administrator prior to activating the HNAS product.

```
[TRCLMT={count|4000}]
                                                   (changed for V2R2M0)
                                                  (changed for V2R4M0)
```

Specifies the number of 16-byte trace entries to be reserved for HNAS diagnostic tracing. HNAS uses the TRCLMT operand to compute the amount of mainframe memory that will be allocated for the system trace table.

You may specify a ***count*** value between 1 and 1048575. If a ***count*** value is not specified or if the specified value is invalid, a default value of **4000** will be used.

Note: TRCLMT= maximum value increase from 32767 to 1048575 was introduced into 240 as Enhancement APAR 2400052.

```
[TRCTRAP=({ALRMLIST=(id1/dat1/sodi,...,idn/datn/sodn),
          RCVLIST=(prot1||pkt1,...,protn||pktn),
          XMTLIST=(prot1||pkt1,...,protn||pktn)
          TRAPACTION={ALL|NONE|
                     ({SUSP|NOSUSP},
                      {SNAP|NOSNAP},
                      EXEC=ddname)})]
                                                     (added for V2R3M0)
                                                   (changed for V2R3M0)
                                                   (changed for V2R4M0)
```

Specifies 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 arguments in order to automatically suspend HNAS internal tracing when any list entry is detected. An action parameter may also be specified to instruct HNAS on what to do when a TRCTRAP 'hit' occurs. This action is in addition to or in place of suspending internal tracing which is the default action.

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.

For **ALRMLIST=**, 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 <u>must</u> 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
012345678901234567890123456789012345678901234567890123456789012345678901234567890l
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 operand was specified:

```
TRCTRAP=(ALRMLIST=(NAS3799I/DIAG=000/195),...)
```

the entire ALRMLIST= operand was rejected:

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 ALRMLIST= parameter 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.

Note: The ALRMLIST= suboperand of the BUILD TRCTRAP= operand should not be confused with the BUILD ALRMFLTR= operand. While they both list alarm IDs in a similar fashion, their functions are completely independent.

For **RCVLIST=** and **XMTLIST=**, you may specify ***prot*i||*pkt*i** values. The protocol value (***prot*i**) must be 00 for XTP or 04 for XOT. The packet data (***pkt*i**) 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 you specify (**n**) 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.

Note: A match on any value in the ALRMLIST=, RCVLIST= or XMTLIST= operands can suspend tracing. The first match in any of these operands is the winner. Multiple operands with multiple values are provided to allow for the greatest flexibility.

The **TRAPACTION=** suboperand allows you to specify what action HNAS is to take when a TRCTRAP ‘hit’ occurs. This 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 and the EXEC command description in the HNAS Console Subsystem Guide.

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).

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 console command is entered.

Note: When multiple trap actions are specified, they are processed in the following order:

1) SUSP
2) SNAP
3) EXEC *ddname*

Examples:

Trap multiple events:

```
BUILD 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 Diagnostic packet received
               XMTLIST=(041BFFFF,   <- any Reset packet sent
                        0413FFFF))  <- any Clear packet sent
```

Trap PAD logon request:

```
BUILD TRCTRAP=(RCVLIST=(04FF6CA063F0, <- l cp (even)
                       04FFEC20E370,  <- l cp (odd)
                       04FF6C206370,  <- l cp (none)
                       04FFCCA0C350,  <- L CP (even)
                       04FF4C2043D0,  <- L CP (odd)
                       04FF4C204350)) <- L CP (none)
```

Trap QLLC Terminate-Self PIU:

```
BUILD TRCTRAP=(RCVLIST=(04FF2C0000FFFFFF0B8000010683)) <- TERM-SELF
```

General Notes:

1. The current trace trapping parameters are not altered when tracing is suspended. To restart trace logging, you must issue the **TRCTRAP RSME** or **TRCALL RSME** console command.

2. The current trace trapping parameters are not altered when a SNAP dump is taken on a TRCTRAP hit. To allow another SNAP dump on a subsequent TRCTRAP hit, you must issue the **TRCTRAP RSMESNAP** console command.

3. To resume a suspended trace, allow another SNAP dump and another command list execution, when all actions are performed on a TRCTRAP hit, you may enter the **RSME**, **RSMESNAP** and **RSMEEXEC** console commands or simply **RSMEALL**.

4. When tracing is suspended, **TRCPRNT** trace entry logging in SYSPRINT is also suspended.

5. 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 MINDATA or TRCVC MAXDATA should be active.

6. **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 arguments 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.

[USSTAB={*name*|ISTINCDT}]

(new for V1R1M4)

Specifies the name of a USS table that will be used as the system default USS table for PCNE and/or PAD application LOGON interpret processing by the MCHSOL routine. For more information on MCHSOL services, please see the description of the APPLNAME= operand of the REMOTE definition statement on page 4-74 of this document.

Note: If both **LOGTAB=** and **USSTAB=** are specified, **LOGTAB=** is processed first by MCHSOL.

The ***name*** value you specify can be any valid assembler language symbol.

If a ***name*** value is not specified or if the specified value is invalid or if the specified value is not found in the library identified by the **VTAMLIB DD** statement, a default value of **ISTINCDT** will be used.

[VCLMT={*count*|<u>+VCLMT</u>}]

Specifies the number of virtual circuit control blocks (VCBs) to be reserved for HNAS operation. One VCB is required for each active XOT or XTP session. For most installations this parameter should be omitted -- the default value (see below) is correct

You may specify a ***count*** value between 1 and 65535 to force the allocation of a specific number of VCBs. This might be done to reduce HNAS storage requirements when the default value is significantly larger than the number of active sessions expected. If a ***count*** value is not specified or if the specified value is invalid, the BUILD VCLMT= value will be set as follows:

To 50 if the sum of all REMOTE VCLMT= values is less than 50.
To the REMOTE sum plus 10% if the REMOTE sum is between 50 and 200.
To the REMOTE sum if the REMOTE sum is 200 or greater.

If the VCLMT value for a TYPE=MCH|XTP REMOTE is omitted, it is computed as the sum of the PVC=, SVC0=, SVC3=, SVC4= and SVC5= *vclmt* values for non-GATEFC or the sum of the LUNAME= *slucnt* values for GATEFC.

If the *count* value specified for the BUILD VCLMT= operand is less than the sum of the REMOTE VCLMT= values, <u>the BUILD *count* is used</u> and the following message is generated:

NAS1101W BUILD VCLMT VALUE OF *count* OVERRIDES REMOTE SUM OF *vclmts* <-WARNING

Warning: Because the BUILD VCLMT= value is less than the required number of virtual circuits needed to support all configured switched SLUs, SVC calls that exceed the BUILD VCLMT= value will be cleared for lack of a VC control block with CAUSE/DIAG=000/130.

Note: Even though the BUILD VCLMT= value is less than the required number of virtual circuits needed to support all configured switched SLUs, PVCs are not affected. A VC control block will always be reserved for all PVCs regardless of the VCLMT= value.

Note:

<u>The **total VCLMT=** count (whether user defined or the computed default) must not exceed the total number of available sockets defined in the stack.</u> Should this condition occur during HNAS activation, the following run time error message will occur for each failed attempt:

NAS2201W CLIENT=*iii.iii.iii.iii*(01998) SOCKID=*xx* PCEID=*yy* NAME=*rmtname*
NAS2201W SOCKET REQUEST FAILED, RC=FFFFFFFF 00000018

If this message is generated with RC=FFFFFFFF 00000018, it indicates that the socket descriptor table is full. This means that there are an insufficient number of sockets defined in the stack. Check **MAXSOCKETS** and **MAXFILEPROC** limits in the active **BPXPRMxx** file in **SYS1.PARMLIB** or contact HNAS support for assistance.

When HNAS connects to the stack it issues an **INITAPI** request. One of the parameters for this request is MAXSOC which is the maximum socket descriptor number (SOCKID) that HNAS expects to use. This value is computed as the sum of the number of LOCAL statements plus the sum of the VCLMT values for all TYPE=XOT REMOTE statements plus 1 for each router connection (i.e., each different router IP address). This sum is then rounded to the next multiple of 32. The computed MAXSOC value should not be greater than the MAX-SOCKETS value on the NETWORK DOMAINNAME(AF_INET) statement and the MAX-FILEPROC statement in BPXPRMxx.

If MAXSOCKETS is set less than the number of sockets in the HNAS CDF, some SOCKET requests will fail. We recommend setting MAXSOCKETS to 10000. This value is global for all APIs that access the stack.

You can also set these parameters dynamically using the **SETOMVS** command although the values will only be in effect for the duration under the current IPL environment (values will return to previous setting when the system is re-IPLed).

LOCAL Definition Statement

The LOCAL definition statement is used to supply the TCP/IP socket structure for HNAS server components. The LOCAL definition statement can be placed anywhere between the BUILD and END statements in the CDF.

You **must** code at least one (1) LOCAL definition statement in a CDF. You **may** code more than one (1) LOCAL definition statement in the same CDF if HNAS is to support multiple servers of the same type or of different types (TYPE=XTP|XOT).

lclname **LOCAL operands** **operands**	**CC/ VRM**	**XTP**	**XOT**	
[INIT=({{ACTIVE|ONLINE}| {IDLE|OFFLINE}} [,DELAYTIME={*minutes*|1}] [,RETRYLMT={*count*|0}])]	230 A/240 C/230			
IPADDR=*a.b.c.d*		R	R	
[OPTIONS=(BALANCERTEIN, BALANCERTEOUT)]	C/230 C/220	I I		
[PORT={*number*| 3065| <- XTP 1998}] <- XOT				
[RTEIN={NONE|({*mchname*| CLEAR|SKIP} [/*dteaddr*{T|S}],...)}] where: *dteaddr* = [@]{d|*}{d|*}...{d|*}	112 N/240 C/230 C/240	I I I I		
[RTEOUT={NONE|(*rmtname* [/*dteaddr*{T|S}] C/210 [/*desctxt*],...)}] where: *dteaddr* = [@]{d|*}{d|*}...{d|*}	112 C/220 C/240 C/240	I I I I		
[SOCLMT={*number*|2000}]	220			
[TCPNAME={*name*|BUILD}]	220			
[TYPE={XTP|XOT}]	112			

lclname

(changed for V2R3M0)

Provides a name for the LOCAL definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF. Normally, ***lclname*** will also appear in the HOME= operand on one or more REMOTE definition statements.

```
[INIT=({ACTIVE|ONLINE}|{IDLE|OFFLINE}}
          [,DELAYTIME={minutes|1}]
          [,RETRYLMT={count|0}])]
                                          (XTP|XOT server)
                                          (new for V2R3M0)
                                      (changed for V2R4M0)
```

Specifies whether the server connection identified by this LOCAL definition statement is to be initially online and made active or offline and left inactive.

ACTIVE or **ONLINE** specifies that the server connection will be initially online and activated. HNAS will automatically attempt to BIND the server's IP address to the TCP/IP stack.

IDLE or **OFFLINE** specifies that the server connection will be initially offline and inactive. HNAS will not attempt to BIND the server's IP address to the TCP/IP stack until the initial state is changed to active via the **VARY LCL ACT|ON** console command. Information on the VARY console command, can be found in the Console book section, VARY Resource State command.

If the INIT= operand is not specified or if the specified value is invalid, a default value of **ACTIVE** will be used.

Note: INIT=ONLINE|OFFLINE support was introduced into 240 as Enhancement APAR 2400014.

Note: The DPCE console command for a LOCAL in the IDLE state will display OFLN as the current socket state.

DELAYTIME={*minutes*|1} specifies that HNAS is to provide a delay of ***minutes*** after a TCPIP BIND failure before the BIND is retried. You may specify a ***minutes*** value between 0 and 62. If 0 is specified, no delay is provided. If the DELAYTIME= suboperand is not specified or if the specified ***minutes*** value is invalid, a default value of **1** minute will be used.

RETRYLMT={*count*|0} specifies that HNAS is to count BIND failures and if the ***count*** limit is reached, to take the LOCAL server offline (IDLE). You may specify a ***count*** value between 0 and 62. If 0 is specified, failed BINDs are retried indefinitely. If the RETRYLMT= suboperand is not specified or if the specified ***count*** value is invalid, a default value of **0** will be used.

Note: If HNAS is stopped and restarted very quickly ***or*** if the same HOME IP address and Port number are being used by another HNAS image that is running under the same stack and the TCPIP SHAREPORT option is ***not*** in effect, the BIND will fail and the following alarm message will be generated.

```
NAS2231W SERVER=010.117.056.170(01998) SOCKID=0000 PCEID=0007 NAME=LXOT
NAS2231W BIND REQUEST FAILED, RC=FFFFFFFF 00000030
```

The BIND is then retried after a delay specified by the DELAYTIME= suboperand of the INIT= operand. The ***errno*** value (EADDRINUSE=30) in the NAS2231W message above indicates that the TCPIP socket is either currently in use or was in use and the socket's 'linger timeout' has not yet expired in the stack.

Prior to V2R3M0 APAR 2300150, when the EADDRINUSE BIND failure was detected, the 'linger timeout' was reset and the DELAYTIME= timer was then started. If the DELAYTIME= operand was omitted, a default delay of 5-minutes was enforced before the BIND was retried. From the operator's perspective, this could appear as if HNAS were hung because the NAS0001I INITIALIZATION COMPLETE message is withheld until all servers come active.

With V2R3M0 APAR 2300150 applied, the socket's 'linger timeout' is reset before the initial BIND is issued for the HOME LOCAL. This should minimize the probability of receiving the EADDRINUSE error condition. If the failure does occur, the smaller DELAYTIME= default should make BIND error recovery operate more quickly. To avoid any delay, you could specify INIT=(ACTIVE,DELAYTIME=0). Caution should be taken, however, when using a zero value for DELAYTIME= and RETRYLMT= because this could cause HNAS to 'thrash' and consume a lot of CPU cycles if a BIND failure is permanent.

Note: When a BIND ends successfully for a server, the following informational alarm message will be issued.

```
NAS2020I SERVER=010.117.056.170(01998) SOCKID=0000 PCEID=0007 NAME=LXOT
NAS2020I SERVER INITIALIZATION COMPLETE
```

Note: If all BINDs fail for a server and a non-zero RETRYLMT= value is specified, the following alarm message is issued:

```
NAS2021W SERVER=010.117.056.170(01998) SOCKID=0000 PCEID=0007 NAME=LXOT
NAS2021W SERVER INITIALIZATION FAILED, LOCAL VARIED OFFLINE
```

When all servers come active or enter the IDLE state (either specifically or after being forced), the NAS0001I INITIALIZATION COMLETE message will be issued.

Note: The NAS0001I alert message will be withheld if a server BIND continues to fail and the RETRYLMT= value is 0.

Note: For additional information for the NAS2020I, NAS2021W, NAS22311W and NAS0001I messages, please refer to the HNAS Messages and Codes document.

IPADDR=*a.b.c.d*

Specifies the IP address for this HNAS server component and is required. This IP address is used by the router network to establish sessions with HNAS via the TCP/IP stack.

The IP address is entered in standard ‘dotted’ notation. The IP address elements ***(a.b.c.d)*** must be given as decimal numbers in the range 0 to 255. If the IP address is not specified or if the specified value is invalid or if the TCP/IP socket (IP address and TCP port number combination) is duplicated elsewhere in the CDF, HNAS will terminate.

Note: Some environments **require** that the HNAS (230 and above) or NASMAIN (220 and below) program be registered under their host resident security subsystem like RACF or Top Secret. This may be required so that the HNAS product can communicate with the host stack TCPIP addresses and processes that are defined in the HNAS Configuration Definition File. Failure to register HNAS with the resident security subsystem in some environments can prevent HNAS from correctly communicating with the TCPIP stack. We suggest that you discuss this potential requirement with your system administrator prior to activating the HNAS product.

The IP address that you specify may be real or virtual. The routers in the network must be configured to route traffic to this IP address which becomes the **HOME** IP address of HNAS and the TCPIP stack.

Real IP addresses are those that define specific network attachments to the host (for example a controller or OSA). The real IP address is the HOME address for the physical attachment.

Virtual IP addresses (VIPA) are those that are used in place of one or more real IP addresses. Thus, VIPA provide an IP address that selects a particular TCP/IP image rather than a specific network attachment. If the mainframe is configured for multiple OSA interfaces, each with its own real IP address, they may all be made to ‘look like’ a single IP address. Virtual IP addressing requires special handling within the TCPIP PROFILE file.

VIPA can be used to run multiple copies of HNAS on the same LPAR. For example, suppose the host mainframe has 4 OSA interfaces, and hence, 4 real IP addresses defined to the TCPIP stack. Suppose also that traffic on the first 2 OSA is to flow to HNAS1 while traffic from the other 2 OSA is to flow to HNAS2. This topology will require the use of 2 VIPA. Assume the following:

OSA1=63.205.100.110
OSA2=63.205.100.111
OSA3=63.205.100.112
OSA4=63.205.100.113
VIPA1=63.205.121.001
VIPA2=63.205.121.002

The TCPIP PROFILE must contain DEVICE and LINK statements for the real and virtual adapters, that is, OSA1, OSA2, OSA3, OSA4, VIPA1 and VIPA2. For example:

```
DEVICE  OSA1       LCS        5F2
LINK    OSA1LINK   ETHERNET     0     OSA1
DEVICE  OSA2       LCS        55A
LINK    OSA2LINK   ETHERNET     0     OSA2
```

```
DEVICE  OSA3        LCS         450
LINK    OSA3LINK    ETHERNET      0    OSA3
DEVICE  OSA4        LCS         490
LINK    OSA4LINK    ETHERNET      0    OSA4
DEVICE  VIPA1       VIRTUAL       0
LINK    VIPA1LNK    VIRTUAL       0    VIPA1
DEVICE  VIPA2       VIRTUAL       0
LINK    VIPA2LNK    VIRTUAL       0    VIPA2
```

The HOME statement would then be coded as follows:

```
HOME
  63.205.121.001   VIPA1LNK
  63.205.100.110   OSA1LINK
  63.205.100.111   OSA2LINK
  63.205.121.002   VIPA2LNK
  63.205.100.112   OSA3LINK
  63.205.100.113   OSA4LINK
```

Notice that the VIPA links are positioned in front of the real links that they will override. Positioning is very important in the HOME statement. The VIPA link that is closest to the real link that follows it, will provide the substitute IP address.

In addition to the above, you must also specify **SOURCEVIPA** as an option in the ASSORTEDPARM statement (older versions of the TCPIP stack) or the IPCONF statement (recent versions of the TCPIP stack). The SOURCEVIPA option tells the stack to replace the real source IP address with the VIPA in all IP datagram's that originate from the stack.

Now, we are ready to define the VIPA to HNAS1 and HNAS2.

For HNAS1, specify:

```
XOTSRVR  LOCAL TYPE=XOT
               IPADDR=63.205.121.001    VIPA1
               PORT=1998
               :
XTPSRVR  LOCAL TYPE=XTP
               IPADDR=63.205.121.001    VIPA1
               PORT=3065
```

For HNAS2, specify:

```
XOTSRVR  LOCAL TYPE=XOT
               IPADDR=63.205.121.002    VIPA2
               PORT=1998
                :
XTPSRVR  LOCAL TYPE=XTP
               IPADDR=63.205.121.002    VIPA2
               PORT=3065
```

For more information on VIPA, please refer to IBM TCP/IP for MVS Customization and Administration Guide (SC31-7134).

**[OPTIONS=(BALANCERTEIN,
BALANCERTEOUT)]**

(new for V2R2M0)
(change for V2R3M0)

Specifies special processing options for this LOCAL definition statement.

BALANCERTEIN specifies that RTEIN= operand entries with special DTE|DCE address subgrouping are to be used in a way (round robin fashion) that balances the number of calls sent to HNAS MCH entries. For additional information, please see RTEIN= operand description. TYPE=XOT must also be specified.

BALANCERTEOUT specifies that RTEOUT= operand entries with no DTE addresses are to be used in a way that balances the number of calls sent to remote routers. For additional information, please see RTEOUT= operand description. TYPE=XOT must also be specified.

[PORT={*number*|3065|1998}]

Specifies the TCP port number for this HNAS server component.

You may specify a ***number*** value between 0 and 65533 (65534 and 65535 are reserved). Currently, ***number*** values of 3065 (XTP) and 1998 (XOT) are the only values allowed. If a ***number*** value is not specified or if the specified value is invalid, a default value of **3065** will be used for an XTP server and **1998** will be used for an XOT server. These default port numbers are the ‘well known’ TCP port numbers for XTP and XOT peers, respectively.

**[RTEIN={NONE|({*mchname*|CLEAR|SKIP}
[/*dteaddr*{T|S}],...)}]**

(XOT server only)
(new for V1R1M2)
(changed for V2R1M0)
(changed for V2R3M0)
(changed for V2R4M0)

Specifies routing information for inbound XOT client connections in the form of a mapping list that identifies (1) the name of a TYPE=MCH REMOTE definition statement (***mchname***) to which the connections will be directed or an action (**CLEAR|SKIP**) that will be performed on a DTE address match and (2) a partial or complete DTE address (***dteaddr***) that controls the mapping of DTE addresses to a specific MCH or action for **SVCs**.

Note: PVCs do not use the RTEIN= operand for inbound connections. The PVC Setup packet that HNAS receives from the router contains the target MCH name for the PVC. This name is configured in the router.

NONE specifies that **no inbound routing** will be provided. All inbound calls will be rejected. Specifying **RTEIN=NONE** instead of omitting the **RTEIN=** operand will prevent a configuration warning message from being generated.

The ***mchname*** value you specify identifies a logical MCH that supplies session control parameters and defines host application access for inbound XOT client connections. The name you specify for ***mchname*** must appear in the name field of a TYPE=MCH REMOTE definition statement that appears elsewhere in the CDF. The ***mchname*** values can be any valid assembler language symbol. If an ***mchname*** value is not specified or if the specified value does not identify a TYPE=MCH REMOTE, it is ignored.

Specify **CLEAR** instead of an *mchname* value to force the inbound call to be cleared when a *dteaddr* match occurs.

Specify **SKIP** instead of an *mchname* value to force the inbound call to be ignored when a *dteaddr* match occurs.

The ***dteaddr*** value you specify may be up to fifteen (15) decimal digits and represents the first n-characters of DTE addresses that will be routed for inbound XOT client connections.

<u>You may also specify filtering characters for the ***dteaddr*** value</u>.

@ as the <u>first</u> *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. At most 14 digits may follow the @ (if a 15 digit match is required, code a 15 digit address). If @ is coded without a following digit (RTEIN=(...,MCH1/@,...)) then no match occurs (the RTEIN position is skipped).

***** within the *dteaddr* is treated as a wildcard character which matches any digit in the called or calling DTE address at the same position as the *.

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**.

Examples:

12**56 will allow DTE addresses of 120056, 120156, 121056, 120256, etc. to be considered a match.
@12**56 will allow DTE addresses ending with 12xx56 to be considered a match (x=0 through 9).

If a ***dteaddr*** value is not specified or if the specified value is invalid, it is ignored.

The ***dteaddr*** follower is a direction indicator that indicates which DTE address from the inbound Call Request packet will be used to match the ***dteaddr*** value. **T** (for target) indicates that the ***called*** DTE address is to be used. **S** (for source) indicates that the ***calling*** DTE address is to be used. The compare is performed left to right for the length of the ***dteaddr*** value only. Thus, a single ***dteaddr*** value can be used to map or filter a collection of DTE addresses (***dteaddr*=**1234 will match call request packet addresses of 1234, 12345, etc.).

The first omitted ***dteaddr*** or the first ***dteaddr*** value that yields a match causes the corresponding MCH named ***mchname*** to be used to set session parameters and provide host application access or the associated **CLEAR|SKIP** action to be taken. If no match occurs for any ***dteaddr*** value in the RTEIN= operand list, the call is cleared.

You may specify from 1 to 1023 RTEIN= list entries.

If the **RTEIN=** operand is omitted, a default value of **NONE** will be used.

With OPTIONS=BALANCERTEIN specified the selection of an MCH for an inbound call operates as follows:

1) The RTEIN= list is processed against the called and/or calling DTE addresses in the call request packet. The MCH address in each RTEIN= entry that could be used for the call is recorded in an internal table. A RTEIN= entry 'could be used' if there is a calling or called address match. A RTEIN= entry with no dte address is treated as the end of the list (even if there are more entries) and the MCH it names does participate in the round robin logic.

2a) If no RTEIN= entries are selected by the RTEIN= scan and RTEIN= does not end with a no DTE address entry then the inbound call is cleared with DIAG=202.

2b) If no RTEIN= entries are selected by the RTEIN= scan and RTEIN= ends with a no DTE address entry then the TYPE=MCH REMOTE addressed by the no DTE address entry is used for the call.

2c) If one or more RTEIN= entries are selected (tabled) by the RTEIN= scan then the MCHs addressed by the table are searched to locate an MCH with a round robin marker (a flag in the MCH control block). If no marker is found the first MCH in the table is marked and used for the call. If a marker is found then it is moved to the next MCH addressed by the table (possibly after wrapping to the first table entry) and that MCH is used for the call.

Examples with OPTIONS=BALANCERTEIN assumed:

```
RTEIN=(MCH1/7777,MCH2/7777,MCH3/7777,MCH4)
```

Calls with a called DTE address starting with 7777 will be routed in round robin fashion to MCH1, MCH2 or MCH3. All other calls go to MCH4.

```
RTEIN=(MCH1/7777S,MCH2/7777S,MCH3/7777S,MCH4)
```

Same as above except selection of MCH1/2/3 is based on the calling DTE address in the call request packet instead of the called DTE address.

```
RTEIN=(MCH1/7777,MCH1/6666,MCH2/7777,MCH2/6666,MCH3/7777,MCH3/6666,MCH4)
```

Calls with a DTE address starting with 7777 or 6666 will be routed in a round robin fashion to MCH1, MCH2 or MCH3. The first 3 calls with any combination of the 7777 or 6666 called addresses will be routed to MCH1, MCH2 and MCH3 (first call to MCH1, second call to MCH2, etc.).

[RTEOUT={NONE|(*rmtname*[/*dteaddr*{T|S}][/*desctxt*],...)}]

(XOT server only)
(new for V1R1M2)
(changed for V2R1M0)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Specifies routing information for outbound XOT client connections in the form of a mapping list that identifies (1) the name of a TYPE=XOT REMOTE definition statement (***rmtname***) to which the connections will be directed and (2) a partial or complete DTE address (***dteaddr***) that controls the mapping of DTE addresses to TCP/IP sockets for XOT client connections for **SVCs**.

When HNAS is attempting to place a call the RTEOUT= list is processed from left to right in order to locate routers that the call request packet should be sent to. If an outbound call fails (clear or no response from the remote) then the call is attempted using subsequent RTEOUT= entries with matching DTE addresses. The call does not fail until the end of the RTEOUT= list is reached. If the call does fail an UNBIND is used to terminate the PLU session.

Note: The following resources or processes do not use RTEOUT= filtering for HNAS initiated (outbound) connections:

PVCs do not use the RTEOUT= operand for outbound connections. The PVC= operand on the initiating MCH identifies the REMOTE that represents the router that will receive the HNAS PVC Setup packet.

TAP=nn TAPping XOT Call Request tapping support does not utilize any RTEOUT= filtering because the TCP/IP session is established directly with the router's IP and PORT addresses. The TAP socket for a router is wired to the REMOTE definition that represents the router.

PING (XOT) console command support also doesn't utilize any RTEOUT= filtering for the same reasons provided under TAP=nn above.

NONE specifies that **no outbound routing** will be provided. All connections must be initiated by the remote routers using inbound Call Request packets (refer to the **RTEIN=** operand description). Specifying **RTEOUT=NONE** instead of omitting the **RTEOUT=** operand will prevent a configuration warning message from being generated.

The ***rmtname*** value you specify identifies a pool of TCP/IP sockets that are used for outbound XOT client connections. The name that you specify for ***rmtname*** must appear in the name field of a TYPE=XOT REMOTE definition statement that appears elsewhere in the CDF. The ***rmtname*** values can be any valid assembler language symbol.

The pool of TCP/IP sockets is created from the IPADDR=, PORT= and VCLMT= values on the named TYPE=XOT REMOTE definition statement. The IPADDR= operand identifies a specific router in the IP network. The PORT= operand identifies the TCP port number for the router's XOT component. The VCLMT= value determines the size of the pool.

For example, IPADDR=192.40.60.4,PORT=1998,VCLMT=4 specifies that for IP address 192.40.60.4, a TCP socket for 4 XOT clients will be reserved in the pool.

If a ***rmtname*** value is not specified or if the specified value does not identify a TYPE=XOT REMOTE, it is ignored.

The ***dteaddr*** value you specify may be up to fifteen (15) decimal digits and represents the first n-characters of DTE addresses that will be routed for outbound XOT client connections.

<u>You may also specify filtering characters for the ***dteaddr*** value</u>.

@ as the <u>first</u> *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* 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**.

The ***dteaddr*** follower is a direction indicator which indicates which DTE address from the outbound Call Request packet will be used to match the ***dteaddr*** value. **T** (for target) indicates that the ***called*** DTE address is to be used. **S** (for source) indicates that the ***calling*** DTE address is to be used.

HNAS will compare the specified ***dteaddr*** value against the ***called*** DTE address of an XOT outbound Call Request packet if the direction indicator is a **T**. HNAS will compare the specified ***dteaddr*** value against the ***calling*** DTE address of an XOT outbound Call Request packet if the direction indicator is an **S**. The compare is performed left to right for the length of the ***dteaddr*** value only. Thus, a single ***dteaddr*** value can be used to map or filter a collection of DTE addresses.

The ***desctxt*** value you specify will take the place of the ***dteaddr*** value in the NAS7717W message if the call to ***dteaddr*** fails. If ***desctxt*** is specified, the DTE ADDR *dteaddr* text in the NAS7717W message will be replaced with DTE IDNT *desctxt*. ***desctxt***, which can be from 1 to 15 characters in length, serves as a text description of the remote DTE. ***desctxt*** may be enclosed in quotes which do not count in the 15 character length limit.

For example: if RTEOUT=(...,XOTCLNT/1234567890/'RMT DTE NYC',...) were specified and the call to DTE 1234567890 failed, DTE IDNT RMT DTE NYC would be displayed in the NAS7717W message instead of DTE ADDR 1234567890.

For GATE (LLC4) sessions, the outbound Call Request packet comes directly from the host CTCP application. The CTCP provides both the ***called*** and ***calling*** DTE addresses as well as facilities and call user data.

For PCNE (LLC0) or PAD (LLC5) sessions, the outbound Call Request packet is created by HNAS when the PLU BINDs the SLU. The ***called*** DTE address comes from SVC0= or SVC5= operand entry, respectively, on the TYPE=MCH REMOTE definition statement that owns the SLU. The ***calling*** DTE address, facilities data and call user data come from DCEADDR=, FAC= and CUD= operands, respectively, on the same TYPE=XTP|MCH REMOTE definition statement or from the TYPE=MXT REMOTE definition statement associated with the SVC0= or SVC5= operand entry.

For QLLC (LLC3) sessions, the outbound Call Request packet is created by HNAS based on the setting of the OPTIONS=CLOTINITYP= parameter for the SPU or if the SPU is identified in an SVC3= operand of an MCH with an outbound ***called*** DTE address (for example, SVC3=(...,*spuname*/*dteaddr*O,...) and OPTIONS=CLOTINITYP=NONE is in effect for the SPU). If OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE is specified, it overrides SVC3= outbound call initiation.

When OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE is specified for an SPU, the ***called*** DTE address comes from the DTEADDR= operand for the SPU. The ***calling*** DTE address, facilities data and call user data come from DCEADDR=, FAC= and CUD= operands, respectively, on the same TYPE=SPU REMOTE definition statement.

When the Call Request results from an outbound SVC3= operand entry, the ***called*** DTE address comes from SVC3= operand entry on the TYPE=MCH REMOTE definition statement that identifies the SPU. The ***calling*** DTE address, facilities data and call user data come from DCEADDR=, FAC= and CUD= operands, respectively, on the same

TYPE=MCH REMOTE definition statement or from the TYPE=MXT REMOTE definition statement associated with the SVC3= operand entry.

The ***dteaddr*** values are selected for comparison left to right in the RTEOUT operand list. When there is an address match, HNAS builds and transmits an X.25 Call Request packet to the router named ***rmtname***:

Router Availability Rules:

1) The router is active. This means that it was configured active via the INIT=ACTIVE (the default) operand or was varied active by the VARY ACT console command if it was configured idle via the INIT=IDLE operand.

2) The router has not be varied idle by the VARY INACT console command.

3) The router has not been taken offline (made temporarily idle) by the HNAS TAP (Keep Alive) mechanism.

4) A port on the router is available to set up a TCP/IP session. This means that the VCLMT= operand value on the TYPE=XOT REMOTE definition for the router has not been reached.

In short, a router is considered unavailable for an outbound Call Request if it has no available sockets or it is in the idle state either because it was configured that way via INIT=IDLE, was manually varied idle via the VARY INACT console command or was detected offline by the TAP Keep Alive mechanism.

If a Call Accept packet is returned the session begins. Starting with V2R3M0, for LLC0, LLC3 and LLC5, if the call fails (the remote router sends a Clear Request or fails to send a Call Accept in which case a timeout will occur), HNAS will continue in the RTEOUT= operand list in an attempt to find another match. This allows a call to be tried at multiple routers for backup purposes. The call will fail (UNBIND sent to PLU) when the end of the RTEOUT= operand list is reached. For LLC4, a call request failure is reported to the CTCP.

Note: If a ***dteaddr*** value results in the selection of a router that has been taken offline by the HNAS TAP= (Keep Alive) mechanism or is marked idle in the CDF (INIT=IDLE), it is considered to be a Call Request failure. For LLC0 and LLC5, HNAS will attempt to retry the call on a backup router if multiple outbound DTE addresses are specified for the SLU in the SVC0= or SVC5= operand list (for example, SVC0=(...,*sluname*/*dteaddr*1-*dteaddr*2-*dteaddr*3**O**,...)). The ***dteaddr*** search is then restarted at the beginning of the RTEOUT= operand list.

The first omitted ***dteaddr*** or the first ***dteaddr*** value that yields a match causes the corresponding TCP/IP socket pool named ***rmtname*** to be used for TCP/IP socket allocation and subsequent XOT client access. If no match occurs for any ***dteaddr*** value in the RTEOUT operand list, the host application's call request is rejected.

If a ***dteaddr*** value is not specified or if the specified value is invalid, it is ignored.

You may specify from 1 to 1023 RTEOUT= list entries.

If the **RTEOUT=** operand is omitted, a default value of **NONE** will be used.

If OPTIONS=BALANCERTEOUT is in effect then consecutive entries with identical DTE addresses in the RTEOUT= list are considered to be a round robin group. Calls will be balanced across the routers in the group. This is illustrated by the following example:

```
LXOT      LOCAL TYPE=XOT IPADDR=xxx.xxx.xxx.xxx
                RTEOUT=(OUT1/123,OUT2/123,OUT3)
                OPTIONS=BALANCERTEOUT


OUT1      REMOTE TYPE=XOT IPADDR=aaa.aaa.aaa.aaa
                 PORT=1998 VCLMT=n1


OUT2      REMOTE TYPE=XOT IPADDR=bbb.bbb.bbb.bbb
                 PORT=1998 VCLMT=n2


OUT3      REMOTE TYPE=XOT IPADDR=ccc.ccc.ccc.ccc
                 PORT=1998 VCLMT=n3
```

When a call is to be made because of a BIND request directed at a callout LLC0 or LLC5 SLU resource or because of a Call Request from a GATE CTCP, HNAS has the ***called*** DTE address it needs to route the call. In order to locate a REMOTE for the Call Request packet, HNAS searches for a match between the DTE addresses in the RTEOUT= operand list and the leading digits of the ***called*** or ***calling*** DTE address.

Because **OPTIONS=BALANCERTEOUT** is specified, the first and second RTEOUT= entries in the above RTEOUT= example form a round robin group (there can be any number of routers in a group). The first call with an address starting with 123 will use the OUT1 router. The second call with an address starting with 123 will use the OUT2 router. The third call with an address starting with 123 will use the OUT1 router. This balances traffic between routers in the group. All entries in the group will be tried in an attempt to establish the call. The BALANCERTEOUT OPTION simply changes the location in the group where the search starts. If the call fails in all entries in the group then it is tried in the remaining RTEOUT= entries that are not in the round group (OUT3, in the above example).

[SOCLMT={*soclmt*|<u>2000</u>}]

(new for V2R2M0)

Specifies the number of client sockets that this HNAS server component can support.

You may specify a ***soclmt*** value between 50 (TCPIP's default) and 65503. If a ***soclmt*** value is not specified a default value of **2000** will be used. If the specified ***soclmt*** value is invalid a cc-8 error condition will be raised and HNAS will terminate after the CDF scan is completed.

If a ***soclmt*** value greater than 2000 is specified, multiple server component Task Information Elements (TIEs) are created for the same server HOME IP address. The number of TIEs created is computed from ***soclmt***/2000+1. HNAS will **LISTEN** for client connections on each TIE that is created. In this case, the TCPIP stack will present connections to HNAS on the TIE having the lease number of active connections in order to load balance the connections across all TIEs for a specific HOME IP address.

Note: When a LOCAL is HOME to multiple TYPE=XOT REMOTEs whose total socket count (sum of the VCLMT= operand values) is greater than 2000, you must specify a SOCLMT= value larger than the VCLMT= sum. In addition to the REMOTE VCLMT= values, a TAP socket is created for each unique REMOTE IPADDR= address (this does not apply to IPADDR=DYNAMIC). The SOCLMT= should always be rounded to the next 2K boundary.

For example, if 3 TYPE=XOT REMOTEs with unique IPADDR= values have VCLMT= values of 2000, 1000 and 1000, the total remote sockets would be 4003 which includes the 3 TAP sockets. So for these 3 TYPE=XOT REMOTEs, the HOME LOCAL should specify SOCLMT=6000.

Note: When the ***soclmt*** value is greater than 2000, the following entry for the PORT statement in TCPIP PROFILE file must be added:

`1998 TCP` ***`hnasname`*** **`NOAUTOLOG SHAREPORT`**

The **SHAREPORT** option is required when reserving a port to be shared across multiple LISTENers on the same interface. This is true whether the same or different HOME IP addresses are used for each listener.

[TCPNAME={*name*|BUILD}]

(new for V2R2M0)

Specifies the name of the TCP/IP address space as it appears to the HNAS address space. This name is used when HNAS establishes its server(s) presence to the TCP/IP program product.

You may specify a ***name*** value of up to 8 non-blank characters. If a ***name*** value is not specified or if the specified value is invalid (text string is too long), a default name from the TCPNAME operand on the BUILD definition statement will be used.

A single copy of HNAS can communicate with more than one TCPIP stack. This can be useful in environments where one stack is used for production and another is used for test and both require HNAS connectivity. The TCPNAME= operand on LOCAL definition statement allows different server components to be connected to different TCPIP stacks.

Note: Some environments **require** that the HNAS (230 and above) or NASMAIN (220 and below) program be registered under their host resident security subsystem like RACF or Top Secret. This may be required so that the HNAS product can communicate with the host stack

TCPIP addresses and processes that are defined in the HNAS Configuration Definition File. Failure to register HNAS with the resident security subsystem in some environments can prevent HNAS from correctly communicating with the TCPIP stack. We suggest that you discuss this potential requirement with your system administrator prior to activating the HNAS product.

[TYPE={<u>XTP</u>|XOT}]

(new for V1R1M2)

Specifies the TYPE of this HNAS server component.

<u>XTP</u> specifies that HNAS will serve XTP client connections via this LOCAL definition.

XOT specifies that HNAS will serve XOT client connections via this LOCAL definition.

If the TYPE operand is not specified or if the specified value is invalid, a default value of **XTP** will be used.

Note: HNAS can act as an XTP and XOT server simultaneously. A single copy of HNAS can support multiple servers of the same type as long as each uses a unique IP address. This means that multiple server HOME IP addresses can be used for the same PORT number. Each server can connect to the same TCPIP stack or different TCPIP stacks as directed by the TCPNAME= operand.

The ***lclname*** value that you specify for the HOME= operand on a REMOTE definition statement wires a client to a server for **shared** connections only (**PORT=1998|3065** specified on the REMOTE definition statement). For **inbound** connections (**PORT=DYNAMIC** specified on the REMOTE definition statement) or **dynamic** connections (**IPADDR=DYNAMIC** specified on the REMOTE definition statement), the HOME server is assigned when the connection is accepted.

Note: When multiple LOCAL definition statements with the same TYPE= and TCPNAME= values are specified, you must specify the **SHAREPORT** option on the **PORT** statement in the **TCPIP PROFILE** file. For example, if you need to configure multiple HNAS XOT servers, the following PORT statement entry should be used:

1998 TCP *hnasname* NOAUTOLOG SHAREPORT

REMOTE Definition Statement

The REMOTE definition statement is used to supply HNAS with a description of a remote router and its X.25 multi-channel link components in the IP network. This description includes the TCP/IP socket structure and X.25 ‘topology’ for the remote router. A REMOTE definition statement can be placed anywhere between the BUILD and END statements in the CDF.

You may code more than one REMOTE definition statement for the same **XTP router** if that router supports more than one physical X.25 access line. In this case, each REMOTE definition statement must specify the same IP address (IPADDR=) and TCP port number (PORT=3065) but different X.25 interface numbers (IFNUM=).

You may code more than one REMOTE definition statement for the same **XOT router**, however, all must specify the same IP address (IPADDR=) and TCP port number (PORT=1998 or PORT=DYNAMIC). Together with the VCLMT= operand value, the IPADDR= and PORT= operand values create pools of TCP sockets for XOT connections from the same router.

For PORT=1998, the collection of sockets is referred to as a ***shared socket pool*** because it can be used for inbound and outbound connections.

For PORT=DYNAMIC, the collection of sockets is referred to as an ***inbound socket pool*** because it can be used for inbound connections only.

You may also create a ***dynamic socket pool*** by specifying a VCLMT= operand value with IPADDR=DYNAMIC and PORT=DYNAMIC. In this case the socket pool can be used for inbound router connections from any router in the network because the IP address and TCP port number are set when the connection is established.

There are restrictions on user assigned TYPE=MCH|XTP REMOTE names, please refer to ***rmtname*** description below for details.

rmtname **REMOTE operands** **operands**	**CC/ VRM**	**XTP**	**XOT**	**MCH**	**MXT**	**SPU**	**DFL**	**DFS**	**DFX**	**DMY**	**SVC**
[APPLNAME=({*pluname*| CONSOLE| MCHSOL| ACQUIRE},...)]	 A/230	 I	I I I I	 I	I I I I	 I 	I I I I	I I I I	I I I I	I I I I	I I I I
[CONNECT={<u>NO</u>|YES|CUD0|SUBD}]			I		I	I	I	I	I	I	I
[CTCP=({*llcid*|*lunmindex*},...)]			I		I	I	I	I	I	I	I
[CUD={<u>NONE</u>| <- XTP|MCH|MXT|DMY|SVC <u>01000000HNASTAP</u>| <------------- XOT <u>C3000000</u>| <------------- SPU [(]*xx...xx*[)]}]	113 C/220 C/230 						I I I I	I I I I	I I I I		

REMOTE

<table>
<tr><th>*rmtname* REMOTE operands

operands</th><th>CC/
VRM</th><th>X
T
P</th><th>X
O
T</th><th>M
C
H</th><th>M
X
T</th><th>S
P
U</th><th>D
F
L</th><th>D
F
S</th><th>D
F
X</th><th>D
M
Y</th><th>S
V
C</th></tr>
<tr><td>[CUD0={ALL|(*value*1,...,*value*n)}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[DCEADDR={NONE|*dceaddr*}]</td><td>113
C/220</td><td></td><td></td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td></td><td></td></tr>
<tr><td>[DFLNAME=(*dflname*1,...,*dfl*namen)]</td><td>N/240</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[DFXNAME=(*dfxname*1,...,*dfx*namen)]</td><td>N/240</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[DTEADDR={NONE|*dteaddr*}]</td><td>230
C/240</td><td>I</td><td></td><td>I</td><td></td><td></td><td>I</td><td>I</td><td>I</td><td></td><td></td></tr>
<tr><td>[FAC={NONE| <- XOT|MXT|SPU|DMY|SVC
0101420909430404| <--------- XTP|MCH
[(]*xx...xx*[)]}]</td><td>113
C/220
C/230</td><td></td><td></td><td></td><td></td><td></td><td>I
I
I</td><td>I
I
I</td><td>I
I
I</td><td></td><td></td></tr>
<tr><td>[GATE={NO|GENERAL}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[HOME=*lclname*]</td><td>220</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[IDBLK=*xxx*]</td><td>220</td><td>I</td><td>I</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[IDNUM=*xxxxx*]</td><td>220</td><td>I</td><td>I</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[IDLETO={*minutes*}]</td><td>230</td><td></td><td>I</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>IFNUM=*number*</td><td></td><td>R</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[INIT={{ACTIVE|ONLINE}|
{IDLE|OFFLINE}}]</td><td>112
A/240</td><td></td><td></td><td></td><td>I
I</td><td></td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td></tr>
<tr><td>IPADDR={DYNAMIC|
a.b.c.d}</td><td>C/210</td><td>I
R</td><td>
R</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
</td><td>I
I</td></tr>
<tr><td>[LLC0={NONE|(*value*1,...,*value*n)}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[LLC3={NONE|(*value*1,...,*value*n)}]</td><td>220</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[LLC4={NONE|(*value*1,...,*value*n)}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[LLC5={NONE|(*value*1,...,*value*n)}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[LOGTAB={*name*|BUILD}]</td><td>210
C/220</td><td></td><td>I</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[LUNAME=(*sluname*[-{A|I}] <--------- MCH|XTP
[/*pluname*[||*]]
[/{*pfxlu*|*rmtname*(4)||F}[-{H|D}]]
[/{*sfxst*|0}]
[/{*slucnt*|1}],...)]</td><td>C/220
C/240
A/240</td><td></td><td>I
I
I
I
I</td><td></td><td>I
I
I
I
I</td><td>I
I
I
I
I</td><td>I
I
I
I
I</td><td>I
I
I
I
I</td><td>I
I
I
I
I</td><td>I
I
I
I
I</td><td>I
I
I
I
I</td></tr>
<tr><td>[LUNAME=(...,*gapcnt*, <------------- SPU
sluname[-{A|I}]
[/*rcvpacnt*]
[/*sndpacnt*]
[/*applid*]
[/*mxtname*],...)]</td><td>C/230
C/240</td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td><td></td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td><td>I
I
I
I
I
I</td></tr>
<tr><td>[LUNAME=(*sluname*[-{A|I}] <------------- DFL
[/*pluname*],...)]</td><td>C/240</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td><td></td><td>I
I</td><td>I
I</td><td>I
I</td><td>I
I</td></tr>
<tr><td>[LUNAME=(*sluname*[-{A|I}] <------------- DFS
/*pluname*
/*idnumm*,...)]</td><td>C/240</td><td>I
I
I</td><td>I
I
I</td><td>I
I
I</td><td>I
I
I</td><td>I
I
I</td><td>I
I
I</td><td></td><td>I
I
I</td><td>I
I
I</td><td>I
I
I</td></tr>
</table>

rmtname REMOTE operands operands	CC/VRM	XTP	XOT	MCH	MXT	SPU	DFL	DFS	DFX	DMY	SVC
[MAXDATA={*bytecnt*|261| <------------ SPU1	220	I	I	I	I		I	I	I	I	
265}] <------------ SPU2		I	I	I	I		I	I	I	I	I
[MBITCHN={NO|YES}]	C/210		I		I	I	I	I	I	I	I
[OPTIONS=(CLOTINITYP={NONE|BIND|TIMER|CONSOLE},	230	I	I	I	I		I	I	I	I	I
CLOTFAILRTYLMT={*count*|3},	230	I	I	I	I		I	I	I	I	I
CLOTCONLMT={*count*|0},	230	I	I	I	I		I	I	I	I	I
CUD0SELECTSLU,	A/240		I			I	I	I	I	I	I
DATAF,	N/240	I	I	I	I	I	I	I		I	I
DATAFAM,	N/240	I	I	I	I	I	I	I		I	I
DELAYBINDRESP,	A/230		I		I	I	I	I	I	I	I
ECHODTEADDR,	220	I	I		I	I	I	I	I	I	I
ECHOFAC,	220	I	I		I	I	I	I	I	I	I
EMSGE,	N/240	I	I	I	I	I	I	I		I	I
IDTST,	A/240	I	I	I	I	I	I	I		I	I
IMS,	N/240	I	I	I	I	I	I	I		I	I
INHIBITBIDREJ,	230		I		I	I	I	I	I	I	I
LCN0USED,	220	I	I		I	I	I	I	I	I	I
LLC0CTCTCHK,	A/230	I	I		I	I	I	I	I	I	I
LLC5CTCPCHK,	A/230	I	I		I	I	I	I	I	I	I
MCHTMR={*seconds*|60},	211		I		I	I	I	I	I	I	I
NOCLOSEONTAPFAILURE,	A/240			I	I	I	I	I	I	I	I
NORTRBIDREJ,	230		I		I	I	I	I	I	I	I
{PRI|SEC|PEER},	220	I	I	I	I		I	I	I	I	I
NRITAB=*name*,	N/240	I	I	I	I	I	I	I		I	I
ONEPIUINB,	230		I		I	I	I	I	I	I	I
PFXDCEADDR,	A/230	I	I		I	I	I	I	I	I	I
PVCRECONTMR={*seconds*|60},	A/240	I	I			I	I	I	I	I	I
PVCSETUPREJ=*code*,	A/240	I	I			I	I	I	I	I	I
PVCSETUPTMR={*seconds*|60},	A/240	I	I			I	I	I	I	I	I
REPDCEADDR,	210	I	I		I	I	I	I	I	I	I
REQSESSDELAY={*seconds*|2},	230		I		I	I	I	I	I	I	I
RESETINO	A/240		I		I	I	I	I	I	I	I
RESIDSTART={*firstid*|pvcct+1},	230		I		I	I	I	I	I	I	I
RETPIU,	N/240	I	I	I	I	I	I	I		I	I
REUSEBUSYSPU,	A/230	I	I	I	I		I	I	I	I	I
STRIPFAC,	210	I	I		I	I	I	I	I	I	I
STRIPRTEIN,	210	I	I		I	I	I	I	I	I	I
SVCCALLTMR={*seconds*|30},	A/240	I	I			I	I	I	I	I	I
TAPWITHCLR,	A/220	I		I	I	I	I	I	I	I	I
TCPRBLMT={*count*|*},	220	7	2	I	I	I	I	I	I	I	I
XID=({NO|STD|TAB|TABSTD|	N/240	I	I	I	I	I	I	I		I	I
(*idnum,idcnt*)})]		I	I	I	I	I	I	I		I	I
[PACE={{*pktcnt*|0}|	112		I	I	I	I	I	I	I	I	I
(*rcvpacnt,sndpacnt*)}]	220	I	I	I	I		I	I	I	I	I
[PAD={NO|INTEG|TRANSP|PACEONLY}]			I		I	I	I	I	I	I	I
[PADPARM={NONE|(*pnum/pval,...*)|			I			I	I	I	I	I	I
1/0,7/21,8/0|			I			I	I	I	I	I	I
1/0,7/2,8/0|			I			I	I	I	I	I	I
3/2,4/0,7/2,13/4}]			I			I	I	I	I	I	I
[PKTSIZ={*bytecnt*|256}]	112 C/220			I	I	I	I	I	I	I	I

REMOTE

<table>
<tr><th>rmtname REMOTE operands
operands</th><th>CC/ VRM</th><th>XTP</th><th>XOT</th><th>MCH</th><th>MXT</th><th>SPU</th><th>DFL</th><th>DFS</th><th>DFX</th><th>DMY</th><th>SVC</th></tr>
<tr><td>[PORT={DYNAMIC|</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>number|</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td></td><td>I</td></tr>
<tr><td>3065| <-------------- XTP</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td></td><td>I</td></tr>
<tr><td>1998}] <-------------- XOT</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td></td><td>I</td></tr>
<tr><td>[PROTOCOL={XTP| <-------------- XTP</td><td>N/240</td><td></td><td></td><td></td><td>I</td><td></td><td></td><td></td><td></td><td></td><td>I</td></tr>
<tr><td>XOT}] <---------- non-XTP</td><td></td><td></td><td></td><td></td><td>I</td><td></td><td></td><td></td><td></td><td></td><td>I</td></tr>
<tr><td>[PVC={NONE|</td><td></td><td></td><td></td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(vclmt},</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[{sluname[-{A|I}]|rmtname(4)||P||i}]</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/{llcid|0}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/{applid|255}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/{lcn|0}]</td><td>112</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>/ifname</td><td>112</td><td>I</td><td>I</td><td>R</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/rmtname]</td><td>112</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/mxtname],...)}]</td><td>A/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[PWPROT={NO|YES|YESWOCC}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[SUBADDR={NO|YES}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[SUBD=(value1,...,valuen)]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[SVC0={NONE|</td><td>C/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(pfxlu,sfxst,vclmt)|</td><td>A/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(vclmt,</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[{sluname[+gluname][-{A|I}]|</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>rmtname(4)||0||i}]</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/{X||<i>idnum1</i>|<i>dteaddr1</i>/<<i>rmtname1</i>>}</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[-{X||<i>idnum2</i>|<i>dteaddr2</i>/<<i>rmtname2</i>>}</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[-{X||<i>idnum3</i>|<i>dteaddr3</i>/<<i>rmtname3</i>>}]]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>||{T|O|I}[{applid|0}]}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/mxtname]</td><td>113</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/cud],...)}]</td><td>220</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[SVC3={NONE|ALLOW|</td><td>C/230</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(vclmt},</td><td></td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[spuname]</td><td>220</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/dteaddr||{I|O}]</td><td>C/230</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/mxtname],...)}]</td><td>220</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[SVC4={NONE|</td><td>C/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(pfxlu,sfxst,vclmt)|</td><td>A/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(vclmt},</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[{sluname[-{A|I}]|rmtname(4)||4||i}],...)}]</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[SVC5={NONE|</td><td>C/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(pfxlu,sfxst,vclmt)|</td><td>A/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>(vclmt,</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[{sluname[+gluname][-{A|I}]|</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>rmtname(4)||5||i}]</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/{X||<i>idnum1</i>|<i>dteaddr1</i>/<<i>rmtname1</i>>}</td><td>C/240</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[-{X||<i>idnum2</i>|<i>dteaddr2</i>/<<i>rmtname2</i>>}</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[-{X||<i>idnum3</i>|<i>dteaddr3</i>/<<i>rmtname3</i>>}]]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>||{T|O|I}[{applid|0}]}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/mxtname]</td><td>113</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[/cud],...)}]</td><td>220</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
</table>

<table>
<tr><th>*rmtname* REMOTE operands

operands</th><th>CC/
VRM</th><th>XTP</th><th>XOT</th><th>MCH</th><th>MXT</th><th>SPU</th><th>DFL</th><th>DFS</th><th>DFX</th><th>DMY</th><th>SVC</th></tr>
<tr><td>[SYSL=({SUBD=*subdval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD0=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD1=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD2=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD3=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD4=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD5=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD6=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD7=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD8=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>CUD9=*cudval/applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>NULL/*applid*|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>DATA=*textstring/applid*},...)]</td><td>211
C/220</td><td>I</td><td>I</td><td>I</td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[TAP={*seconds*|0}]</td><td>C/210
C/220</td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[TRAN={NO|USER|EVEN|ODD|MARK|SPACE|</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>NPSIEVEN|NPSIODD|</td><td>A/230</td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>NPSIMARK|NPSISPACE}]</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[TYPE={XTP|XOT|MCH|MXT|
SPU1|SPU2|SPU|
DFL|DFS|DFX|DMY|SVC}]</td><td>112
C/220
N/240</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>[USSTAB={*name*|BUILD}]</td><td>114
C/220</td><td></td><td>I</td><td></td><td></td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>[VCLMT={*count*|1|</td><td>C/220</td><td></td><td>1</td><td></td><td>I</td><td>I</td><td></td><td></td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>+PVC+SVCi| <-- non-GATEFC MCH</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td><td>I</td></tr>
<tr><td>+LUNAME}] <-- GATEFC|DFL|DFS</td><td></td><td></td><td>I</td><td></td><td>I</td><td>I</td><td></td><td></td><td>I</td><td>I</td><td>I</td></tr>
</table>

rmtname

(changed for V2R3M0)

Provides a name for the REMOTE definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF. Normally, ***rmtname*** will also appear in the RTEIN= and/or RTEOUT= operand on one or more LOCAL definition statements or in various operands on other REMOTE definition statements.

Note: The ***rmtname*** you specify for TYPE=MCH|XTP REMOTE definition statements must be unique in the first 4 character positions because these characters are used to generate default SLU names when entries are omitted from the PVC=, SVC0=, SVC4= and SVC5= operands.

[APPLNAME=({*pluname*|CONSOLE|MCHSOL|ACQUIRE},...)]
(SPU|MCH|XTP client)
(changed for V2R2M0)
(changed for V2R3M0)

Specifies a list non-CTCP application (PLU) names that HNAS can provide access to. APPLNAME= operand connections are simulated LUT1 connections. The APPLNAME= operand is **required** for PCNE (LLC0) and PAD (LLC5) sessions. It is used in conjunction with the SYSL= operand for application selection. If the SYSL= operand is omitted, the first (or only) APPLNAME= operand entry is used as the default application.

The ***pluname*** values you specify identify host applications that are accessed via the non-GATE (PCNE or PAD) function. The ***pluname*** values can be any valid assembler language symbol.

If a ***pluname*** value is specified as **CONSOLE** then selection of that APPLNAME= entry will result in connection to the HNAS Console Subsystem. See Console book section for more information on HNAS console support. **CONSOLE** is invalid with TYPE=SPU.

If a ***pluname*** value is specified as **MCHSOL** then selection of that APPLNAME= entry will result in connection to the HNAS MCH Solicitor. MCHSOL is an HNAS component that provides LOGTAB and/or USSTAB interpret processing. This is required because HNAS LUs do not have an SSCP/SLU session. HNAS LUs operate exclusively within a PLU/SLU session.

In the NPSI world, the SSCP provides LOGTAB and/or USSTAB. In the HNAS world, it is MCHSOL that provides this function. With NPSI MAXOUT= on the PU statement in the Switched Major Node is used to control whether or not the first message from an LLC0/LLC5 device is sent to the SSCP as a logon message. Specifically, MAXOUT=6 keeps NPSI from sending the first message top the SSCP. Other MAXOUT values cause the first message to be sent to the SSCP. Because HNAS is an APPL LU there is no switched major node and no SSCP session. With HNAS, the processing of the first message is controlled by whether or not MCHSOL is selected as a 'PLU' via APPLNAME=.

If a ***pluname*** value is specified as **ACQUIRE** then selection of that APPLNAME= entry (for TYPE=SPU REMOTE definitions only) will result in the ACB for an SLU being opened and a SETLOGON request being passed to VTAM but a REQSESS request being withheld. This operation conditions the SLU to accept an application BIND but does not solicit the BIND. The SLU remains passive. This support is primarily used for printer LUs that are acquired based on need via requests from other LUs. This APPLNAME= operand entry may be selected via a 'hard wired' *applid* value (see LUNAME= operand) or via USSTAB processing.

If a ***pluname*** value is not specified or if the specified value is invalid, the name is ignored.

You may specify from 1 to 64 application names.

Example: `APPLNAME=(TSO,CICS,CONSOLE)`

Note: The CONSOLE and MCHSOL functions are <u>not available</u> to PVC connections.

[CONNECT={NO|YES|CUD0|SUBD}]

(MCH|XTP client)

Specifies how the GATE Fast Connect (GATEFC) function is to be activated.

Note: For all CONNECT= values except NO, GATE=GENERAL must also be specified.

NO specifies that GATE Fast Connect is not used.

If the LUNAME= operand specifies a single CTCP SLU name and the CUD0= and CTCP= operands are omitted, internally CTCP=(0,0,0,...,0,0) and CUD0=(C0,C4,02,...,2F,NULL) are assumed. A GATE (non-GATEFC) session will be established to the CTCP for those DTEs whose **inbound** Call Request packet CUD0 byte appears in the internal CUD0= operand list.

YES specifies that GATE Fast Connect is used to access a single CTCP. In this case, the LUNAME= operand specifies a single CTCP SLU name, The CTCP=, CUD0= and SUBD= operands must be omitted.

CUD0 specifies that GATE Fast Connect is used and access to a particular CTCP is controlled by mapping CUD0 (Call User Data byte 0) values. In this case, the LUNAME= operand specifies a list of CTCP SLU names, the CTCP= operand specifies a list of indices into the LUNAME= operand list and the CUD0= operand specifies a list of CUD0 values. There must be a one-to-one correspondence between the CTCP= and CUD0= operands. The SUBD= operand must be omitted.

If the LUNAME= operand specifies a single CTCP SLU name and the CUD0= and CTCP= operands are omitted, internally CTCP=(0,0,0,...,0,0) and CUD0=(C0,C4,02,...,2F,NULL) are assumed. A GATEFC session will be established to the CTCP for those DTEs whose **inbound** Call Request packet CUD0 byte appears in the internal CUD0= operand list.

Example:
```
GATE=GENERAL                    ; GATE FUNCTION REQUIRED
CONNECT=CUD0                    ; MAP CUD0 TO GATEFC CTCP
CUD0=(00,01,C0,NULL,02)         ; SUPPLY CUD0 VALUES
CTCP=(0,0,1,2,0)                ; SUPPLY LUNAME INDICES
LUNAME=(CTCPSLU0////100,        ; SUPPLY CTCP SLU NAMES/LIMITS
        CTCPSLU1////100,
        CTCPSLU2////10)
```

CTCP application selection for SVCs is accomplished as follows:

1. When an XTP or XOT **inbound** Call Request packet is received, a lookup is performed in the CUD0= operand list using the CUD0 byte from the packet. The CUD0 byte is a binary number.

2. If no match is found, the call is cleared.

3. If a match occurs, the position of the matched CUD0 value in the CUD0= operand list is used as an index into the CTCP= operand list.

4. The CTCP= operand list entry is then used as an index into the LUNAME= operand list. If the selected CTCP SLU is not bound, the call is cleared.

5. In the example above, a CUD0 value 00, 01 or 02 results in the selection of CTCPSLU0, a CUD0 value C0 results in the selection of CTCPSLU1 and a NULL CUD0 value (no call user data at all) results in the selection of CTCPSLU2.

SUBD specifies that GATE Fast Connect is used and access to a particular CTCP is controlled by mapping SUBD (subaddress digits) values. In this case, the LUNAME= operand specifies a list of CTCP SLU names, the CTCP= operand specifies a list of indices into the LUNAME= operand list and the SUBD= operand specifies a list of SUBD values. There must be a one-to-one correspondence between the CTCP= and SUBD= operands. The CUD0= operand must be omitted.

```
Example: GATE=GENERAL                   ; GATE FUNCTION REQUIRED
         CONNECT=SUBD                   ; MAP SUBD TO GATEFC CTCP
         SUBD=(0,1,14,20,23)            ; SUPPLY SUBD VALUES
         CTCP=(0,0,1,2,0)               ; SUPPLY LUNAME INDICES
         LUNAME=(CTCPSLU0////100,       ; SUPPLY CTCP SLU NAMES/LIMITS
                 CTCPSLU1////100,
                 CTCPSLU2////10)
```

CTCP application selection for SVCs is accomplished as follows:

1. When an XTP or XOT **inbound** Call Request packet is received, a lookup is performed in the SUBD operand list using the SUBD byte from the packet. The SUBD byte is a packed decimal number.

2. If no match is found, the call is cleared.

3. If a match occurs, the position of the matched SUBD value in the SUBD= operand list is used as an index into the CTCP= operand list.

4. The CTCP= operand list entry is then used as an index into the LUNAME= operand list. If the selected CTCP SLU is not bound, the call is cleared.

5. In the example above, a SUBD value 0, 1 or 23 results in the selection of CTCPSLU0, a SUBD value 14 results in the selection of CTCPSLU1 and a SUBD value of 20 results in the selection of CTCPSLU2.

If the CONNECT= operand is not specified or if the specified value is invalid, a default value of **NO** will be used.

[CTCP=({*llcid*|*lunmindex*},...)]

(MCH|XTP client)
(changed for V2R2M0)

(changed for V2R4M0)

Specifies a mapping list that is used to establish Logical Line Control (LLC) or CTCP selection for switched virtual circuits (SVCs).

For GATE=GENERAL,CONNECT=NO (standard GATE), the CTCP= operand specifies a list of LLC identifiers and/or indices (relative to zero) into the LUNAME= operand list. When ***llcid*** values are used, the APPLNAME= operand specifies a list of non-CTCP application names and the SYSL= operand specifies a list of SUBD and/or CUD*i* mapping values. Together, these operands provide a 'system select' capability for PCNE and PAD sessions. See the following LLC mapping example.

Note: For standard GATE, when LLC4 is established via subaddress selection (SUBADDR=YES,LLC4=(*list*)), the VC is always connected to the first CTCP from the LUNAME= operand. The CUD0= and CTCP= operands are not used.

Note: For standard GATE, when subaddress selection (SUBADDR=YES,LLC*i*=(*list*)) is in effect but no match results in any LLC*i*=(*list*) operand, LLC type and CTCP selection are performed using the CUD0 byte in conjunction with the CUD0= and CTCP= operands.

For GATE=GENERAL,CONNECT=CUD0|SUBD (GATE Fast Connect), the CTCP= operand specifies a list of indices into LUNAME= operand list only.

Note: For GATEFC, CONNECT=SUBD and CONNECT=CUD0 are mutually exclusive and SUBADDR=YES,LLC*i*=(*list*) are invalid (no LLC selection by subaddress because the MCH is dedicated to Fast Connect resources).

You may specify ***llcid*** values of 80 for LLC0 (PCNE), 83 for LLC3 (QLLC) or 85 for LLC5 (PAD).

For Datafono support, you may specify ***llcid*** values between 100 and 120 (inclusive). For additional information on this operand, see Datafono TYPE=MCH REMOTE Definition description on page 4-163.

You may specify ***lunmindex*** values between 0 and the LUNAME= operand entry count minus one (N'LUNAME-1). In the latter case, a default ***llcid*** value of 84 for LLC4 (GATE) is assumed.

If an ***llcid*** or ***lunmindex*** value is not specified or if the specified value is invalid, it will be ignored.

You may specify from 1 to 254 CTCP= operand values.

Example:
```
GATE=GENERAL                    ; GATE FUNCTION REQUIRED
CONNECT=NO                      ; MAP CUD0 TO CTCP ENTRY
PAD=TRANSP                      ; XPAD SUPPORT REQUIRED
CUD0=(00,01,C0,NULL,02,C1,C5)   ; SUPPLY CUD0 VALUES
CTCP=(0,0,1,2,0,80,85)          ; SUPPLY LUNAME AND LLC IDS
LUNAME=(CTCPSLU0,               ; SUPPLY CTCP SLU NAMES
        CTCPSLU1,
```

```
                 CTCPSLU2)
         SYSL=(CUD4=41/0,CUD4=42/1,     ; SUPPLY APPL MAPPING
               CUD0=C5/2,SUBD=99/2)
         APPLNAME=(TSO,CICS,CONSOLE)    ; IDENTIFY NON-CTCP APPS
```

CTCP or non-CTCP application selection for SVCs is accomplished as follows (SUBADDRESS= not coded):

1. When an **inbound** Call Request packet is received, a VC control block is allocated and a lookup is performed in the CUD0= operand list using the CUD0 byte from the packet. The CUD0 byte is a binary number.

2. If no match is found, the call is cleared.

3. If a match occurs, the position of the matched CUD0 value in the CUD0= operand list is used as an index into the CTCP= operand list.

4. If the CTCP= operand list entry is a value less than or equal to 27, it is used as an index into the LUNAME= operand list for CTCP application selection. If the selected CTCP SLU is not bound, the call is cleared.

5. If the CTCP= operand list entry is a value greater than 27, it is used as a Logical Line Control identifier. A value of 80 specifies LLC0 (PCNE), a value of 83 specifies LLC3 (QLLC) and a value of 85 specifies LLC5 (PAD).

6. When the CUD0/CTCP mapping yields a value of 83 (LLC3), HNAS returns a Call Accept packet followed by a QLLC XID Request packet to the remote DTE. The SPU for the QLLC VC will be allocated when the XID Response is received based on IDBLK/IDNUM matching. After the SPU is allocated, the SPU and all its associated SLUs are activated. Host application (PLU) selection is performed during the first data exchange with the SLUs. For more information on PLU selection, please see SYSL=DATA=, LOGTAB= and USSTAB= operands.

7. When the CUD0/CTCP mapping yields a value of 80 (LLC0) or 85 (LLC5), an SLU is allocated from the SVC0= or SVC5= operand list, respectively.

   If the corresponding SVC0= or SVC5= operand entry has an associated ***applid*** value, it is used to select an application from the APPLNAME= operand list. If the selected host application is not active (REQSESS rejected) or if the Console Subsystem is selected and it is unavailable, the call is cleared.

   If the corresponding SVC0= or SVC5= operand entry does not have an associated ***applid*** value, the SYSL= operand is then used for application selection. The SUBD= and/or CUD*i*= suboperands of the SYSL= operand specify mapping values and indices into the APPLNAME= operand list. If no match occurs in SYSL=, the first APPLNAME= entry is used as a default. If the selected host application is not active (REQSESS rejected) or if the Console Subsystem is selected and it is unavailable, the call is cleared.

8. In the example above, a CUD0 value 00, 01 or 02 results in the selection of CTCPSLU0, a CUD0 value C0 results in the selection of CTCPSLU1 and a NULL CUD0 value (no call

user data at all) results in the selection of CTCPSLU2. A CUD0 value of C1 or C5 results in LLC the selection. For LLC selection, a CUD4 value of 41 results in the selection of TSO, a CUD4 value of 42 results in the selection of CICS and a CUD0 value of C5 or a SUBD value of 99 results in the selection of the HNAS Console Subsystem.

Examples for CONNECT=CUD0 and CONNECT=SUBD processing can be found on page 4-75 of this document.

```
[CUD={NONE|01000000C8D5C1E2E3C1D7|
           C3000000|
           [(]xx...xx[)]}]
                              (SVC|DMY|SPU|MXT|MCH|XOT|XTP client)
                                                  (new for V1R1M3)
                                              (changed for V2R2M0)
                                              (changed for V2R3M0)
                                              (changed for V2R4M0)
```

Specifies either **NONE** or a string of hexadecimal digits that will be placed in the Call User Data field of all non-GATE XTP or XOT **outbound** Call Request packets that originate from this REMOTE definition.

NONE specifies that no Call User Data will be supplied by HNAS in outbound Call Request packets. Specifying **CUD=NONE** instead of omitting the **CUD=** operand will prevent a configuration warning message from being generated.

You may specify an ***xx...xx*** value as a hexadecimal string without the framing characters “X” up to 63 bytes in length.

Note: HNAS does not check the CUD= operand data for standardized X.25 network 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. Although up to 63 bytes of CUD= operand 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.

Note: For TYPE=XOT REMOTE definition statements, a TAP= operand value must also be specified.

Note: For TYPE=MCH|XTP REMOTE definition statements, at least one (1) ***called*** DTE address (‘O’ for outbound) must also be specified in the SVC0= or SVC5= operand. A non-GATE outbound Call Request packet is created when a PLU BIND request is directed at a outbound SVC0= or SVC5= operand entry.

Note: For TYPE=SPU REMOTE definition statements, CLOTINITYP=BIND|TIMER|CONSOLE must also be specified in the OPTIONS= operand.

Note: For TYPE=DMY REMOTE definition statements, the CUD= value you specify is used in the Call Request packet that results from the PING *dmyname* console command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

Note: For a TYPE=MXT|SVC REMOTE definition statement that is referenced in an SVC0= or SVC5= operand for outbound or twoway usage (O|T), the CUD= operand value you specify is used as call user data in the outbound Call Request packet that results when the SLU is bound. If omitted on a referenced TYPE=MXT|SVC REMOTE definition statement, the default value is taken from the CUD= operand on the root TYPE=MCH|XTP REMOTE definition statement. If a CUD= operand value is specified for a referenced TYPE=MXT|SVC REMOTE definition statement, it always takes precedence over the CUD= operand value from the root TYPE=MCH|XTP REMOTE definition statement (see Call Request Parameter Hierarchy Matrix on page 4-133 for more information).

If an ***xx...xx*** value is not specified or if the specified value is invalid, a default value of **01000000C8D5C1E2E3C1D7** (**01000000+HNASTAP**) will be used for TYPE=XOT REMOTE definition statements, **C3000000** will be used for TYPE=SPU definition statements and **NONE** will be used for TYPE=MXT REMOTE definition statements. A default value is not set for TYPE=MCH|XTP REMOTE definition statements.

If a ***cud*** suboperand value is specified for a SVC0= or SVC5= operand entry, it always take precedence over the CUD= operand value from a referenced TYPE=MXT REMOTE definition statement as well as the root TYPE=MCH|XTP REMOTE definition statement unless a TYPE=MXT|SVC REMOTE name (<*rmtname*>) is specified in place of a DTE address. In this case, the CUD= operand on the referenced TYPE=MXT|SVC takes precedence (see Call Request Parameter Hierarchy Matrix on page 4-133 for more information). If no ***cud*** suboperand is specified and the CUD= operand is also omitted, a default value of C0000000 will be used for PCNE (LLC0) and a default value of 01000000 will be used for PAD (LLC5).

Note: A default value for the CUD= operand is no longer set for a TYPE=MCH|XTP REMOTE definition statement. This change was implemented so that the correct default CUD can be set for PCNE and PAD calls. In earlier releases, a default CUD of 01000000 was set for both PCNE and PAD. This forced the remote DTE to provide PAD services for a PCNE call. By not setting a default for the CUD= operand, the correct default CUD can be set for PCNE and PAD VCs when the outbound Call Request packet is built.

Note: When converting from NPSI to HNAS, the HNAS CUD= parameter provides the function of the USRFILD parameter.

[CUD0={ALL|(*value*1,...,*value*n,NULL)}]

(MCH|XTP client)

Specifies how the first byte of call user data (CUD0) in an **inbound** call request packet is used to establish LLC type and/or CTCP selection. GATE=GENERAL is required.

CUD0= omitted
LLC4 (GATE) and the first CTCP (see LUNAME=) are selected if there is no call user data or if CUD0 is X'00', X'C4' or X'02 through X'2F'. Omit the CTCP= operand.

CUD0=ALL
LLC4 (GATE) and the first CTCP (see LUNAME=) are selected when CUD0 is X'02' or X'C4'. LLC4 (GATE) and the second CTCP (see LUNAME=) are selected when CUD0 is omitted or equal to X'00' or in the range X'03' through X'2F'. Omit the CTCP= operand.

CUD0=(set of hex CUD0 values and/or NULL)
The set of numbers is used to select a CTCP and/or LLC type in conjunction with the CTCP= operand (which is required and must have the same number of elements as CUD0=).

Valid CUD0= operand values are hex 0 through FF (without framing characters X'') and NULL.

You may specify up to 254 CUD0= operand list entries. The number of entries in the CUD0= and CTCP= operands must be the same.

Specified values override the default values associated with CUD0=ALL. Unspecified values take the default CUD0=ALL values.

Note: When CUD0=ALL is coded or CUD0= is omitted, X'C0' or X'CC' as CUD0 selects LLC0 (PCNE) and X'01', X'41', X'51' or X'81' as CUD0 selects LLC5 (PAD).

See CTCP= operand for an example.

[DCEADDR={NONE|*dceaddr*}]

(SVC|DMY|SPU|MXT|MCH|XOT|XTP client)
(new for V1R1M3)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

Specifies either **NONE** or a string of decimal digits that will be placed in the ***calling*** DTE address field of all non-GATE or XOT TAP **outbound** Call Request packets that originate from this REMOTE definition statement. For GATE sessions, the ***calling*** DTE address will be set if **OPTIONS=PFXDCEADDR|REPDCEADDR** is coded and the CTCP did not specify a calling address in it's call request packet.

NONE specifies that no ***calling*** DTE address will be supplied by HNAS in outbound Call Request packets. Specifying **DCEADDR=NONE** instead of omitting the **DCEADDR=** operand will prevent a configuration warning message from being generated.

You may specify a ***dceaddr*** value as a decimal string up to 15 digits in length.

Note: HNAS does not check the DCEADDR= operand data for standardized X.25 network values. The only requirement is that the data must be decimal and the number of digits must not exceed 15.

Note: For TYPE=XOT REMOTE definition statements, a TAP= operand value must also be specified.

Note: For TYPE=MCH|XTP REMOTE definition statements, at least one (1) ***called*** DTE address (with ‘O’ for outbound or ‘T’ for twoway) must also be specified in the SVC0= or SVC5= operand. A non-GATE outbound Call Request packet is created when a PLU BIND request is directed at a outbound SVC0= or SVC5= operand entry.

Note: For TYPE=SPU REMOTE definition statements, CLOTINITYP=BIND|TIMER|CONSOLE must also be specified in the OPTIONS= operand.

Note: For TYPE=DMY REMOTE definition statements, the DCEADDR= value you specify is used as the *calling* DTE address in the Call Request packet that results from the PING *dmyname* console command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

Note: For a TYPE=MXT|SVC REMOTE definition statement that is referenced in an SVC0= or SVC5= operand for outbound or twoway usage (O|T), the DCEADDR= operand value you specify is used as the *calling* DTE address in the outbound Call Request packet that results when the SLU is bound. If omitted on a referenced TYPE=MXT|SVC REMOTE definition statement, the default value is taken from the DCEADDR= operand on the root TYPE=MCH|XTP REMOTE definition statement. If a DCEADDR= operand value is specified for a referenced TYPE=MXT|SVC REMOTE definition statement, it always takes precedence over the DCEADDR= operand value from the root TYPE=MCH|XTP REMOTE definition statement (see Call Request Parameter Hierarchy Matrix on page 4-133 for more information).

If a ***dceaddr*** value is not specified or if the specified value is invalid, a default value of **NONE** will be used for TYPE=XTP|XOT|MCH|SPU|DMY|SVC REMOTE definition statements.

[DFLNAME=(*dflname*1,...,*dflname*n)]

(Datafono MCH client)
(new for V2R4M0)

Specifies the names of up to 64 TYPE=DFL REMOTE definition statements. These statements provide a POOL of pseudo-leased SLU resources that can be used by Datafono calls routed to this MCH. A leased SLU resource is used when the TYPE=DFX REMOTE for the session has the XID=NO option specified (see OPTIONS= for TYPE=DFX REMOTE). If this operand is omitted, pseudo-leased resources are not supported on this MCH. For additional information on this operand, see Datafono TYPE=MCH REMOTE Definition description on page 4-163.

[DFXNAME=(*dfxname*1,...,*dfxname*n)]

(Datafono MCH client)
(new for V2R4M0)

Specifies the names of up to 21 TYPE=DFX REMOTE definition statements that are used to provide Datafono session parameters. If the CTCP value is 100, the TYPE=DFX REMOTE named *dfxname*1 is used. If the CTCP value is 101, the TYPE=DFX REMOTE named *dfxname*2 is used and so on. If the CTCP value is 120 the 21st TYPE=DFX REMOTE name is used. If there is no DFXNAME= entry for a given CTCP value, the Datafono call is cleared. For additional information on this operand, see Datafono TYPE=MCH REMOTE Definition description on page 4-163.

[DTEADDR={NONE|*dteaddr*}]

(SVC|DMY|SPU|MXT|XOT client)
(new for V2R3M0)
(changed for V2R4M0)

Specifies either **NONE** or a string of decimal digits that will be placed in the ***called*** DTE address field of all QLLC or XOT TAP Call Request packets that originate from this REMOTE definition statement.

NONE specifies that no ***called*** DTE address will be supplied by HNAS in outbound Call Request packets. Specifying **DTEADDR=NONE** instead of omitting the **DTEADDR=** operand will prevent a configuration warning message from being generated.

You may specify a ***dteaddr*** value as a decimal string up to 15 digits in length.

Note: HNAS does not check the DTEADDR= operand data for standardized X.25 network values. The only requirement is that the data must be decimal and the number of digits must not exceed 15.

Note: For TYPE=XOT REMOTE definition statements, a TAP= operand value must also be specified.

Note: For TYPE=SPU REMOTE definition statements, CLOTINITYP=BIND|TIMER|CONSOLE must also be specified in the OPTIONS= operand.

Note: For TYPE=DMY REMOTE definition statements, the DTEADDR= value you specify is used as the *called* DTE address in the Call Request packet that results from the PING *dmyname* console command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

Note: For a TYPE=MXT|SVC REMOTE definition statement that is referenced in an SVC0= or SVC5= operand for outbound or twoway usage (O|T), the DTEADDR= operand value you specify is used as the *called* DTE address in the outbound Call Request packet that results when the SLU is bound. For a TYPE=MXT|SVC REMOTE definition statement that is referenced in an SVC0= or SVC5= operand for inbound usage (I), the DTEADDR= operand value you specify is used to match the *calling* DTE address in an inbound Call Request packet. If a match occurs, the SLU is allocated for the inbound call. If DTEADDR= is omitted on a referenced TYPE=MXT|SVC REMOTE definition statement, the named REMOTE is bypassed in the SVC0 or SVC5 operand list (see Call Request Parameter Hierarchy Matrix on page 4-133 for more information).

If a ***dteaddr*** value is not specified or if the specified value is invalid, a default value of **NONE** will be used.

```
[FAC={NONE|0101420909430404|
        [(]xx...xx[)]}]
                           (SVC|DMY|SPU|MXT|MCH|XOT|XTP client)
                                               (new for V1R1M3)
                                           (changed for V2R2M0)
                                           (changed for V2R3M0)
                                           (changed for V2R4M0)
```

Specifies either **NONE** or a string of hexadecimal digits that will be placed in the facilities field of all non-GATE XTP or XOT **outbound** Call Request packets that originate from this MCH.

NONE specifies that no facilities data will be supplied by HNAS in outbound Call Request packets. Specifying **FAC=NONE** instead of omitting the **FAC=** operand will prevent a configuration warning message from being generated.

You may specify an ***xx...xx*** value as a hexadecimal string without the framing characters “X” up to 63 bytes in length.

The facilities field provides for Call Request packet facility codes and facility parameters such as: Normal Charge or Reverse Charge, Normal Priority or High Priority (010n identifies the various combinations for these 4 requests), Throughput Class Negotiation (02nn), Closed User Group (03nn), Maximum Packet Size (420n0n) and Window Size Request (430n0n) values for the requested network. The XOT specification requires that window size and packet size be provided in all XOT initiated Call Request packets.

Note: HNAS does not check the FAC= operand data for standardized X.25 network 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.

Note: For TYPE=XOT REMOTE definition statements, a TAP= operand value must also be specified.

Note: For TYPE=MCH|XTP REMOTE definitions statements, at least one (1) ***called*** DTE address (‘O’ for outbound) must also be specified in the SVC0= or SVC5= operand. A non-GATE outbound Call Request packet is created when a PLU BIND request is directed at a outbound SVC0= or SVC5= operand entry.

Note: For TYPE=SPU REMOTE definition statements, CLOTINITYP=BIND|TIMER|CONSOLE must also be specified in the OPTIONS= operand.

Note: For TYPE=DMY REMOTE definition statements, the FAC= value you specify is used in the Call Request packet that results from the PING *dmyname* console command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

Note: For a TYPE=MXT|SVC REMOTE definition statement that is referenced in an SVC0= or SVC5= operand for outbound or twoway usage (O|T), the FAC= operand value you specify is used as facilities data in the outbound Call Request packet that results when the SLU is bound. If omitted on a referenced TYPE=MXT|SVC REMOTE definition statement, the default value is taken from the FAC= operand on the root TYPE=MCH|XTP REMOTE definition statement. If a FAC= operand value is specified for a referenced TYPE=MXT|SVC REMOTE definition statement, it always takes precedence over the FAC= operand value from the root TYPE=MCH|XTP REMOTE definition statement (see Call Request Parameter Hierarchy Matrix on page 4-133 for more information).

If an ***xx...xx*** value is not specified or if the specified value is invalid, a default value of **0101420909430404** will be used for TYPE=MCH|XTP REMOTE definition statements while **NONE** will be used for all others.

Note: 0101 => allow reverse charging of call with normal priority
420909 => set packet size to 512 (2**9)
430404 => set window size to 4

Note: When converting from NPSI to HNAS, the packet size facility (42) and window size facility (43) can be created from the MAXPKTL= and VWINDOW= operands, respectively, on the appropriate X25.VCCPT macro. Alternatively, the entire FAC= operand value can be taken from the appropriate X25.OUFT macro. The appropriate X25.VCCPT or X25.OUFT macro is identified by the VCCINDX= or OUFINDX= operands, respectively, on the X25.VC macro that is associated with the X25.MCH which this HNAS logical MCH is emulating.

[GATE={<u>NO</u>|GENERAL}]

(MCH|XTP client)

Specifies whether Generalized Access Transport Extension (GATE) functions will be activated for XTP or XOT connections.

<u>NO</u> specifies that no GATE functions will be used. Host access will be provided by the PCNE or PAD functions only.

GENERAL specifies that GATE functions will be allowed. CONNECT=, CTCP=, CUD0=, LLC0=, LLC3=, LLC4=, LLC5=, LUNAME=, SUBD= and SUBADDR= are associated operands.

If the GATE= operand is not specified or if the specified value is invalid, a default value of **NO** will be used.

[HOME=*lclname*]

(MCH|XOT|XTP client)
(new for V2R2M0)

Specifies the name of a the HOME server component for this REMOTE definition statement. The name that you specify for ***lclname*** must appear in the name field of a LOCAL definition statement that appears elsewhere in the CDF. The ***lclname*** value you specify can be any valid assembler language symbol.

For TYPE=XTP|XOT REMOTE definitions statements, the HOME= operand is used to identify a LOCAL definition statement that will provide a source IP address and TCP socket for outbound connections.

For TYPE=MCH REMOTE definition statements, the HOME= operand is used to identify a LOCAL definition statement that will provide the RTEOUT= operand mapping for outbound connections.

If a ***lclname*** value is not specified or if the specified value is invalid or if the named LOCAL definition statement is not found in the CDF, a default server component will be assigned to the client. A default server will be assigned by the configuration for a logical MCH, XTP client or XOT ***shared socket pool*** (REMOTE with IPADDR=*a*.*b*.*c*.*d* and PORT=1998). A default server will be assigned at connection establishment time for an XOT ***inbound socket pool*** (REMOTE with IPADDR=*a*.*b*.*c*.*d* and PORT=DYNAMIC) or an XOT ***dynamic socket pool*** (REMOTE with IPADDR=DYNAMIC and PORT=DYNAMIC).

If a default server is assigned by the configuration process, it is based on matching the REMOTE TYPE= operand with the LOCAL TYPE= operand (REMOTE TYPE=MCH is treated as TYPE=XOT for the purpose of finding a default server). If multiple servers of the same type are defined, the default server assigned by the configuration may not be correct. In this case, you should explicitly specify the correct ***lclname*** value for the HOME= operand.

Note: When a LOCAL is HOME to multiple TYPE=XOT REMOTEs whose total socket count (sum of the VCLMT= operand values) is greater than 2000, you must specify a SOCLMT= value larger than the VCLMT= sum. In addition to the REMOTE VCLMT= values, a TAP socket is created for each unique REMOTE IPADDR= address (this does not apply to IPADDR=DYNAMIC). The SOCLMT= should always be rounded to the next 2K boundary.

For example, if 3 TYPE=XOT REMOTEs with unique IPADDR= values have VCLMT= values of 2000, 1000 and 1000, the total remote sockets would be 4003 which includes the 3 TAP sockets. So for these 3 TYPE=XOT REMOTEs, the HOME LOCAL should specify SOCLMT=6000.

For additional information on HOME server usage, please see the description of the TYPE= operand for the LOCAL definition statement.

[IDBLK=*xxx*]

(SPU client only)
(new for V2R2M0)

Specifies the ID block number that is carried in the QLLC XID response from a real SNA DTE. The IDBLK= operand value you specify will be used in conjunction with the IDNUM= operand value to establish a connection between the remote PU in the network and the logical SPU identified by this TYPE=SPU REMOTE definition statement.

You may specify a ***xxx*** value as a hexadecimal string without the framing characters “X’’ up to 3 digits in length.

Note: HNAS does not check the IDBLK= operand 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.

If a ***xxx*** value is not specified or if the specified value is invalid, HNAS will terminate.

[IDNUM=*xxxxx*]

(SPU client only)
(new for V2R2M0)

Specifies the ID number that is carried in the QLLC XID response from a real SNA DTE. The IDNUM= operand value you specify will be used in conjunction with the IDBLK= operand value to establish a connection between the remote PU in the network and the logical SPU identified by this TYPE=SPU REMOTE definition statement.

You may specify a ***xxxxx*** value as a hexadecimal string without the framing characters “X’’ up to 5 digits in length.

Note: HNAS does not check the IDNUM= operand 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.

If a ***xxxxx*** value is not specified or if the specified value is invalid, HNAS will terminate.

[IDLETO=*minutes*]

(SPU|MXT|MCH|XTP client)
(new for V2R3M0)

Specifies an inactivity timeout value for HNAS virtual circuit connections associated with the REMOTE statement. PVC and remote console sessions are exempt from inactivity timeouts. If a VC does not receive or send data in the specified ***minutes*** interval then the virtual circuit call is cleared with cause=198. A zero value indicates that inactivity timeouts are not provided. The maximum value is 255.

The sample below shows the HNAS logic for assigning the IDLE timeout interval for an LLC5/LLC0 session.

```
IDLETO ASSIGNMENT FOR LLC0/LLC5 RESOURCES
        BUILD     IDLETO= ...
MCH1  REMOTE TYPE=MCH IDLETO=
                     SVC0=(n,LU1/<MXT1>-<MXT2>-<MXT3>T/MXT4,...) ...
```

INBOUND LLC0/LLC5 call processing:
An LLC0 call is assigned to HNAS SLU LU1 if the calling DTE address matches the DCEADDR= coded on the MXT1, MXT2 or MXT3 REMOTEs (not shown). The MXT with the

address ‘hit’ provides the IDLETO for the inbound call. If IDLETO is omitted on this MXT then the IDLETO= value is taken from MXT4 (if coded), MCH1 (if coded) or BUILD -- in that order.

OUTBOUND LLC0/LLC5 call processing:
When LU1 is BOUND (acquired by the PLU) HNAS will try outbound calls using the DTEADDR values coded in MXT1, MXT2 and MXT3. If the call is successful (call accept received) then MXT used for the call provides the IDLETO value. If using MXT1 the MXT1 IDLETO value is used (if coded). If IDLETO is omitted on this MXT then the IDLETO= value is taken from MXT4 (if coded), MCH1 (if coded) or BUILD -- in that order.

QLLC (LLC3) sessions require an SPU REMOTE which provides the LU configuration for the remote PU. The IDLETO parameter on the SPU provides the idle timeout value for the VC session. If the parameter is not coded on the SPU then values are taken from the MCH (if coded) or from BUILD. For inbound calls a calling DTE address match in the SVC3= parameter on an MCH can be used to locate the SPU. When this is method is used the SVC3 MXT pointer provides the primary IDLETO value.

For GATE (LLC4) calls the MCH and BUILD IDLETO values are used. The SVC4= parameter does not allow MXT pointers.

IFNUM=*number*

(XTP client only)

Specifies an X.25 multi-channel link (MCH) interface number for this remote XTP router and is required.

The interface number identifies a specific physical X.25 link on the router. The identifier is carried in all XTP packets that are transferred between HNAS and the router.

You may specify a ***number*** value between 0 and 255. If a ***number*** value is not specified or if the specified value is invalid or is defined more than once for this router, HNAS will terminate.

[INIT={<u>ACTIVE</u>|<u>ONLINE</u>}|{IDLE|OFFLINE}}]

(SPU|MCH|XOT|XTP client)
(new for V1R1M2)
(changed for V2R4M0)

Specifies whether the resource identified by this REMOTE definition statement is to be initially online and allowed to activate or offline and inactive.

<u>ACTIVE</u> or **<u>ONLINE</u>**

For TYPE=XOT|XTP specifies that the router connection will be initially online and will be allowed to become active. HNAS will automatically attempt to make contact with the target router and will allow inbound connections from the router.

For TYPE=SPU specifies that the SPU is available for normal operation (inbound and outbound QLLC traffic possible).

For TYPE=MCH specifies that MCH resources are available for normal operation.

IDLE or **OFFLINE**

For TYPE=XOT|XTP specifies that the router connection will be initially offline. HNAS will not attempt to contact the target router and will not allow inbound connections from the router.

For TYPE=SPU specifies that the SPU is not available for use. The ACBs for LUs associated with the SPU will not be opened so VTAM will see the resources as inactive.

For TYPE=MCH specifies that MCH resources are not available. The MCH is skipped if it is selected by RTEIN= processing. The ACBs for LUs associated with the MCH are not opened so VTAM will see the resources as inactive. GATE control sessions are not activated, PVC SETUP packets are not generated or accepted.

The resource IDLE state can be changed to active via the **VARY RMT ACT|ON** console command. Information on the VARY console command, can be found in the Console book section, VARY Resource State command.

If the INIT= operand is not specified or if the specified value is invalid, a default value of **ACTIVE** will be used.

Note: INIT=ONLINE|OFFLINE support was introduced into 240 as Enhancement APAR 2400014.

Note: The DPCE console command for a REMOTE in the IDLE state will display OFLN as the current socket state.

IPADDR={DYNAMIC|*a.b.c.d*}

(DMY|XOT|XTP client)
(changed for V2R1M0)
(changed for V2R4M0)

Specifies the IP address for this REMOTE definition statement and is required. This IP address is used to request a session with a router when HNAS acts as session initiator or to accept a session with a router when HNAS acts as session listener.

For XTP, session initiation observes the following rule: the ‘peer’ with the lower IP address initiates the connection. For example, if HNAS has an IP address of 192.40.60.2 and a router has an IP address of 192.40.60.1, the router would assume the role of session initiator and HNAS would wait (listen) for the router to connect. If the IP addresses were reversed, the opposite would occur, that is, HNAS would initiate the connection and the router would wait.

For XOT, **DYNAMIC** specifies that the source IP address is to be resolved at connection establishment time. The real IP address is set when a router, acting as session initiator, sends a SYN request and HNAS, acting as session listener, issues a TCPIP stack ACCEPT request to accept the connection.

Dynamic IP address assignment is **valid for Cisco (XOT) routers only**. Dynamic IP address assignment eliminates the need to pre-define routers to HNAS when only inbound connections will be used. When dynamic IP address assignment is in affect for a router, shoulder tapping (KEEP ALIVE) processing is inhibited. PORT=DYNAMIC must also be specified.

For **dynamic IP address** assignment, specifying **IPADDR=DYNAMIC, PORT=DYNAMIC, VCLMT=n** for a TYPE=XOT REMOTE definition statement creates a pool of **n**-sockets that are used for **inbound** connections only.

A single REMOTE definition statement with a **dynamic IP address** is all that is required to create a ***dynamic socket pool***. The VCLMT= operand value specifies the number of sockets in the pool. If multiple REMOTE definitions with a dynamic IP address are provided, the sum of their VCLMT= operand values will be used to build the dynamic socket pool.

For a **fixed IP address** assignment, session initiation is controlled by the **PORT=** operand value as described in the following paragraphs.

A single REMOTE definition statement with a **fixed IP address** and **PORT=DYNAMIC** is all that is required to create an ***inbound socket pool*** for a specific router. The VCLMT= operand value specifies the number of sockets in the pool. If multiple REMOTE definitions with the same fixed IP address and PORT=DYNAMIC are provided, the sum of their VCLMT= operand values will be used to build the inbound socket pool for the target router.

A single TYPE=XOT REMOTE definition statement with a **fixed IP address** and **PORT=1998** is all that is required to create an ***shared socket pool*** (inbound and outbound) for a specific router. The VCLMT= operand value specifies the number of sockets in the pool. If multiple REMOTE definitions with the same fixed IP address and PORT=1998 are provided, the sum of their VCLMT= operand values will be used to build the shared socket pool for the target router. A socket from the pool is allocated to an activating session on a first come, first served basis. The same socket can be used for an inbound connection at one time then an outbound connection at another time.

If you are planning to use all three types of socket pools (**dynamic**, **inbound** and **shared**), they should be specified in this order within the CDF. This will allow sockets to be allocated from the ***dynamic socket pool*** before any are allocated from the ***inbound socket pool*** and from the ***inbound socket pool*** before any are allocated from the ***shared socket pool***. This ensures that more sockets from the ***shared socket pool*** remain available for outbound connections.

The IP address is entered in standard ‘dotted’ notation. The IP address elements **(*a.b.c.d*)** must be given as decimal numbers in the range 0 to 255.

Note: For TYPE=XTP|XOT REMOTE definition statements, if the IP address is omitted or is invalid or is duplicated elsewhere in the CDF, HNAS will terminate.

Note: For TYPE=DMY REMOTE definition statements, the IPADDR= value you specify is used to identify the router that will be the target of the PING *dmyname* console command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

General notes for socket pools:

1. When a ***dynamic socket pool*** appears before an ***inbound socket pool*** or ***shared socket pool*** (see below) in the CDF, it will be used for socket allocation even if the source IP address from the inbound connection matches the fixed IP address from those other pools. When a ***dynamic socket pool*** appears after an ***inbound socket pool*** or ***shared socket pool*** in the CDF, it will be used for socket allocation only when all the sockets in those other pools are in use or when the source IP address from the inbound connection does match the fixed IP address from those other pools.

2. When an ***inbound socket pool*** appears before a ***dynamic socket pool*** or ***shared socket pool*** in the CDF, it will be used for socket allocation only if the source IP address from the inbound connection matches the fixed IP address from the inbound socket pool. When an ***inbound socket pool*** appears after a ***dynamic socket pool*** or ***shared socket pool*** in the CDF, it will be used for socket allocation only when all the sockets in those other pools are in use.

3. When an ***shared socket pool*** appears before a ***dynamic socket pool*** or ***inbound socket pool*** in the CDF, it will be used for socket allocation only if the source IP address from the inbound connection matches the fixed IP address from the shared socket pool. When an ***shared socket pool*** appears after a ***dynamic socket pool*** or ***inbound socket pool*** in the CDF, it will be used for socket allocation only when all the sockets in those other pools are in use.

[LLC0={NONE|(*value*1,...,*value*n)}]
[LLC3={NONE|(*value*1,...,*value*n)}]
[LLC4={NONE|(*value*1,...,*value*n)}]
[LLC5={NONE|(*value*1,...,*value*n)}]

(MCH|XTP client)

Specifies either **NONE** or a list of subaddress values that are used to select the Logical Line Control (LLC) type. Normally when an inbound call is received HNAS determines the LLC type from Call User Data Byte 0 (CUD0) in the call request packet (see CUD0 to LLC Type Table in Chapter 3). When GATE=GENERAL and SUBADDR=YES are coded, the above operands can be used to set the LLC type from the subaddress digit in the call request packet. This digit is defined as the last digit in the ***called*** DTE address field. Subaddress LLC selection is performed using values from the above operands before CUD0 processing. If a digit does not appear as an LLC0=, LLC3=, LLC4= or LLC5= operand value then LLC and CTCP selection are performed using the CUD0 value and information provided by the CTCP= and CUD0= operands.

If LLC4 (GATE) is selected by subaddress then the first CTCP identified by LUNAME= is used for the session.

NONE specifies that the corresponding LLCi cannot be selected by subaddress.

***value*i** specifies a subaddress digit value (0 through 9) that selects an LLC type. The values coded must be unique across all LLCi= operands.

After the LLC type has been set by subaddress (or CUD0 if the subaddress does not appear in an LLCi= operand) HNAS proceeds with session setup using the SVC0=, SVC3=, SVC4= or SVC5= operands to obtain an LU and PLU for the session.

Example: LLC0=(3,4,9)

Specifies that if the subaddress digit is 3, 4, or 9 then the LLC type is 0 (PCNE).

[LOGTAB={*name*|BUILD}]

(SPU|MCH|MXT|XTP client)
(new for V2R1M0)
(changed for V2R2M0)

Specifies the name of a LOGON table that will be used for PCNE and/or PAD application LOGON interpret processing by the MCHSOL routine. MCHSOL processing is activated when MCHSOL is selected from the APPLNAME= operand. For more information on MCHSOL services, see the description of the APPLNAME= operand on page 4-74 of this document.

Note: When both **LOGTAB=** and **USSTAB=** are specified, **LOGTAB=** is processed first by MCHSOL.

The ***name*** value you specify can be any valid assembler language symbol. If a ***name*** value is not specified or if the specified value is invalid or if the specified value is not found in the library identified by the **VTAMLIB DD** statement, the default value will be taken from the **LOGTAB=** operand on the **BUILD** definition statement.

[LUNAME=(...)]

(DFS|DFL|SPU|MCH|XTP client)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

This parameter supplies LU names for GATE control sessions, GATE Fast Connect Data Sessions, remote QLLC configurations and Datafono leased and switched SLU pools. Each SLU name must appear on an APPL statement in the HNAS VTAM Application Major Node File (AMNF). All LU names must be valid assembler language names. The valid operands for this parameter depend on the type of REMOTE being defined.

For non-Fast Connect GATE (TYPE=MCH or TYPE=XTP REMOTEs)

LUNAME=(*sluname*[-{A|I}][/*pluname*],...)

sluname identifies a GATE control session SLU name. Up to 28 may be specified.

All HNAS SLU names must appear in an APPL statement in an active application major node file (AMNF).

-{A|I} specifies the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When HNAS activates and the SLU initial state is active, the ACB for the GATE control session is opened. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for GATE processing. If the control session is idle then there can be no inbound or outbound GATE calls.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

[/*pluname*] specifies an optional PLU name for ***sluname***. When this parameter is coded, HNAS will issue a REQSESS macro to request a BIND from the named PLU. If this operand is omitted The PLU must acquire the HNAS SLU to start the control session. Most installations code this operand to ensure that the control session starts whether HNAS or the CTCP is activated first.

For Fast Connect (FC) GATE (TYPE=MCH or TYPE=XTP REMOTEs)

```
LUNAME=(sluname[-{A|I}]
          [/pluname[||*]]
          [/pfxlu|rmtname(4)||F[-{H|D}]
          [/sfxst|0
          [/slucnt|1],...)
```

sluname specifies a GATE FC control session SLU name. Up to 28 may be coded.

All HNAS SLU names must appear in an APPL statement in an active application major node file (AMNF).

-{A|I} specifies the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When HNAS activates and the SLU initial state is active, the ACBs for the GATE FC control session and the GATE FC data sessions are opened. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the control and data session SLUs for GATE FC processing.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

[/*pluname*[*]] specifies an optional PLU name for ***sluname***. When this parameter is coded, HNAS will issue a REQSESS macro to request a BIND from the named PLU to start the

GATE control session. If this operand is omitted the PLU must acquire the HNAS SLU to start the control session. The optional '*' follower directs HNAS to also issue a REQSESS macro to request a bind to start the data session LUs defined by this operand. FC Data Session LUs, like GATE control session LUs, are always active (HNAS ACB open). If the '*' is omitted the data session LUs must be acquired by the PLU. Most installations code this operand to ensure that the control session and data sessions are started whether HNAS or the CTCP is activated first.

[/*pfxlu* | *rmtname*(4)||F] specifies the leading characters of the data session SLU names generated for this FC control session LU. Up to 5 characters may be coded for ***pfxlu***. If the parameter is omitted the default LU name prefix is the first 4 characters of the REMOTE's name (# characters used as filler if the REMOTE's name has fewer than 4 characters) followed by the character 'F'.

-{H|D} specifies the data session SLU name generation type (-H (or omitted) for ascending hexadecimal names or -D for ascending decimal names). -H (hex) is the default name generation type.

Note: *pfxlu*-{H|D} support was introduced into 240 as Enhancement APAR 2400035.

When *pfxlu*-H or *pfxlu* (without the -H) is specified:

[/*sfxst* | 0] specifies a hexadecimal number (without the framing characters X'') between 0 and FFF when the ***pfxlu*** length is 5, between 0 and FFFF when ***pfxlu*** length is 4, ..., between 0 and FFFFFFF when the ***pfxlu*** length is 1. If ***sfxst*** is omitted (//), a default value of 0 will be used. The suffix number is stepped by 1 for each SLU name generated.

When *pfxlu*-D is specified:

[/*sfxst* | 0] specifies a decimal number between 0 and 999 when the ***pfxlu*** length is 5, between 0 and 9999 when ***pfxlu*** length is 4, ..., between 0 and 9999999 when the ***pfxlu*** length is 1. If ***sfxst*** is omitted (//), a default value of 0 will be used. The suffix number is stepped by 1 for each SLU name generated.

[/*slucnt* | 1] specifies the number of GATE FC data session LUs to be generated for this control session LU. The ***slucnt*** value must be a decimal number such that when the *sfxst* value is incremented, it will not cause an SLU name to exceed 8 characters. The SLU names that HNAS generates from the ***pfxlu***, ***sfxst*** and ***slucnt*** values will always be 8-characters in length with zero (0) pad characters added as required between the last ***pfxlu*** character and the suffix value. If the parameter is omitted a count of 1 is used.

HNAS generates eight (8) character terminal SLU names from the ***pfxlu***, ***sfxst*** and ***slucnt*** operands. For example, if **/*pfxlu*/*sfxst*/*slucnt*** are given as **/XRU/9/5**, the data session LU names would be XRU00009, XRU0000A, XRU0000B, XRU0000C and XRU0000D.

For TYPE=SPU REMOTEs

```
LUNAME=(...,gapcnt,
            sluname[-{A|I}]
            [/rcvpacnt]
            [/sndpacnt]
            [/applid]
            [/mxtname],...)
```

LUNAME= identifies the HNAS application SLUs (APPL statements in the AMNF) used for sessions with SLUs on the REMOTE SPU and defines the relationship between devices on the remote SPU (addressed by a LOCADDR value) and the HNAS application SLUs. The LUNAME= operand also provides session control parameters for the HNAS application SLUs.

gapcnt is a decimal number specifying a gap in the local address (LOCADDR) sequence at the remote SPU device. The *gapcnt* parameter is useful if there are large gaps in the local addresses used at the remote SPU.

When a *gapcnt* is not specified, the LOCADDR value for an SLU is assigned based on the position of the *sluname* in the LUNAME= operand list.

For LUNAME=(*sluname*1,*sluname*2,...), *sluname*1 would use LOCADDR=1, *sluname*2 would use LOCADDR=2, and so on. LOCADDR=0 is reserved for the SPU itself.

When a *gapcnt* is specified, the LOCADDR for the SLUs that follow *gapcnt* start at previous LOCADDR value plus the *gapcnt* value plus 1.

For LUNAME=(*sluname*1,*sluname*2,**10**,*sluname*13,*sluname*14,...), *sluname*1 would use LOCADDR=1, *sluname*2 would use LOCADDR=2, *sluname*13 would use LOCADDR=13, *sluname*14 would use LOCADDR=14 and so on.

If a *gapcnt* is the first LUNAME= operand entry, the LOCADDR for the SLUs that follow *gapcnt* would simply be the *gapcnt* value plus 1.

For LUNAME=(**10**,*sluname*11,*sluname12*,*sluname*13,*sluname*14,...), *sluname*12 would use LOCADDR=11, *sluname*12 would use LOCADDR=12, *sluname*13 would use LOCADDR=13, *sluname*14 would use LOCADDR=14 and so on.

sluname specifies the name of an HNAS application SLU. The position of the name in the LUNAME= operand list provides the local address at the remote. The first *sluname* is for LOCADDR=1, the second *sluname* is for LOCADDR=2, etc. If consecutive LOCADDR values are not used at the remote, missing addresses must be represented by consecutive commas (,,,) or by coding a decimal gap count value (see ***gapcnt*** above).

All HNAS SLU names must appear in an APPL statement in an active application major node file (AMNF).

-{A|I} specifies the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When an SPU connection is established and the SLU initial state is active, an ACTLU is sent to the SLU. When the SLU transmits an INITSELF request, the

ACB for the SLU is opened to allow communication with the host application. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for host communication.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

[/*rcvpacnt*] specifies a receive pacing count that will be installed in the BIND image sent by HNAS to the remote if the SLU receive pacing count in the BIND image sent to HNAS from the PLU had a zero receive pacing count (BINRPACE field). Values between 0 and 253 are valid. If ***rcvpacnt*** is not specified the *rcvpacnt* value from the PACE= parameter is used.

[/*sndpacnt*] specifies a send pacing count that will be installed in the BIND image sent by HNAS to the remote if the SLU send pacing count in the BIND image sent to HNAS from the PLU had a zero send pacing count (BINAPACE field). Values between 0 and 253 are valid. If ***sndpacnt*** is not specified the *sndpacnt* value from the PACE= parameter is used.

[/*applid*] specifies an index to an entry in the APPLNAME= operand list. Values from 0 (selects first APPLNAME= entry) to the number of entries in the APPLNAME= list minus 1 are valid.

If ***applid*** is specified, when the remote SLU activates ('Power On' received in ACTLU response or in a NOTIFY request received from the remote) a RESQSESS macro is issued to ask the PLU identified by ***applid*** for a BIND to start the session. If a value is not specified (//), the remote SLU operator will need to select a PLU using USSTAB processing (MCHSOL) or SYSL=DATA processing. If there are remote devices that only operate when they are acquired by a PLU then ***applid*** should reference an APPLNAME= entry with the reserved name ACQUIRE. When this is done, HNAS will wait for the SLU to be acquired (bound).

[/*mxtname*] specifies an the name of a TYPE=MXT REMOTE statement used to override operands on the TYPE=SPU REMOTE. If ***sluname*** has an associated ***mxtname*** operand then the USSTAB= from the TYPE=MXT REMOTE overrides the TYPE=SPU REMOTE's USSTAB= operand. Any valid assembler symbol name may be used.

LUNAME=(D1,,,,D2) is equivalent to LUNAME=(D1,3,D2).

Both examples specify that there are two devices at the remote which are addressed via local addresses of 1 and 5. The PLU communicates with these devices using LU names D1 and D2 (HNAS application LUs).

For TYPE=DFL REMOTEs

LUNAME=(...,*sluname*[-{A|I}]
[/*pluname*],...)

sluname identifies a Datafono pseudo-leased terminal device. Up to 64 names may be specified in the LUNAME= parameter.

All HNAS SLU names must appear in an APPL statement in an active application major node file (AMNF).

-{A|I} specifies the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When the SLU initial state is active, the SLU is conditioned to accept Datafono connections. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for Datafono connections.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

[/*pluname*] specifies an optional PLU name for ***sluname***. When this parameter is coded, HNAS will issue a REQSESS macro to request a BIND from the named PLU. If this operand is omitted, the PLU must acquire the HNAS SLU to start the control session.

For additional information on this operand, see Datafono TYPE=DFL REMOTE Definition description on page 4-166.

For TYPE=DFS REMOTEs

```
LUNAME=(...,sluname[-{A|I}]
             [/pluname]
             [/idnum],...)
```

sluname identifies a Datafono switched terminal device. Up to 64 names may be specified in the LUNAME= parameter.

All HNAS SLU names must appear in an APPL statement in an active application major node file (AMNF). Any number of DFS REMOTEs may be coded (there is a system wide limit of 510 REMOTEs). All DFS switched resources are placed in a common pool used for all Datafono calls.

-{A|I} specifies the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When the SLU initial state is active, the SLU is conditioned to accept Datafono connections. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for Datafono connections.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

[/*pluname*] specifies an optional PLU name for ***sluname***. When this parameter is coded, HNAS will issue a REQSESS macro to request a BIND from the named PLU. If this operand is omitted, the PLU must acquire the HNAS SLU to start the control session.

[/*idnum*] specifies an optional ID number PLU name for ***sluname*. *idnum*** should be coded as a 5 digit hexadecimal number for CUD IDNUM matching.

For additional information on this operand, see Datafono TYPE=DFS REMOTE Definition description on page 4-167.

[MAXDATA={*bytecnt*|261|265}]

(SPU client only)
(new for V2R2M0)

Specifies the maximum PIU size for this remote SPU. The specified value is used only when a maximum PIU size value is not provided in the XID response that HNAS receives from the SPU when the VC is connection is first established.

You may specify a ***bytecnt*** value between 128 and 8192 which includes the Transmission Header (TH), Request/Response Header (RH) and the Request/Response Unit (RU). The RU represents the payload in a PIU and is variable in length. The RH identifies the PIU type and has a fixed length of three (3) bytes. The TH contains routing information and has a length of six (6) bytes for a PUT2 and two (2) bytes for a PUT1. If a ***bytecnt*** value is not specified or if the specified value is invalid, a default value of **261** (bytes) will be used for PUT1 (TYPE=SPU1) and **265** will be used for PUT2 (TYPE=SPU2|SPU).

The ***bytecnt*** value is used to control TH segmentation. When a host PIU is greater than the ***bytecnt*** value it is broken up into Basic Information Units (BIUs) whose length is ***bytecnt****.* The first BIU contains the original TH, RH and a portion of the RU. Subsequent BIUs contain a copy of the TH and more of the RU. This continues until the entire RU has been transmitted. TH segmentation is required to prevent overrunning the SPU receive buffer.

[MBITCHN={NO|YES}]

(MCH|XTP client)
(changed for V2R1M0)

Specifies how RU chaining is to be supported on inbound and outbound flows. The NPSI default for MBITCHN is NO. For virtually all applications MBITCHN=YES is correct. Accordingly the HNAS default for this operand is YES.

For XOT:

NO specifies that an inbound (to HNAS) M-bit chain will be delivered to the PLU as a series of only-in-chain RUs. Each RU in an outbound RU chain will be sent as an M-bit packet sequence. NO (which prevents the PLU from seeing the M-bit chain as a complete RU) is rarely correct. When converting a NPSI GATE data session LU definition with MBITCHN=NO to an HNAS GATE data session LU, be sure to code MBITCHN=YES in the HNAS definition (or let the parameter default). See Host Programming (SC30-3502) section “RU chaining for Virtual Circuits Working in GATE and Transparent PAD” for details.

YES specifies that an inbound M-bit chain of packets will be delivered to the PLU as an RU chain (FIC/MIC/LIC) sequence. An outbound RU chain will be sent as a single M-bit chain of X.25 packets. YES is correct for virtually all PLUs.

For XTP, the protocol does not include M bits. On the inbound (to HNAS) flow the router collects an M-bit chain and sends a single XTP message to HNAS to represent the chain. On the outbound flow the router converts an XTP message to an M-bit chain.

NO specifies that an inbound (to HNAS) XTP data message will be delivered to the PLU as a series of OIC RUs. Each RU in an outbound RU chain will be sent as an XTP message.

YES specifies that an inbound XTP data message will be delivered to the PLU as an RU chain (FIC/MIC/LIC). An outbound RU chain will be delivered in a single XTP message.

For GATE and Transparent PAD sessions, MBITCHN=YES is treated as MBITCHN=NO for inbound traffic. That is, each XTP or XOT packet will be passed to the PLU as a series of OIC RUs based on the inbound RU size established by the PLU's BIND request (MAXRU).

For XOT, starting with the V1R1M4 release of HNAS, each complete M-bit chained packet sequence for GATE and PAD sessions will be delivered to the host as a single OIC RU when the SLU is bound with a zero (0) RU size (unlimited). This makes HNAS consistent with NPSI.

For XOT, starting with the V2R1M0 release of HNAS, if the MBITCHN operand is not specified or if the specified value is invalid, a default value of **YES** will be used. Earlier releases of HNAS defaulted to **NO**.

```
[OPTIONS=(CLOTINITYP={NONE|BIND|TIMER|CONSOLE},
          CLOTFAILRTYLMT={count|3},
          CLOTCONLMT={count|0},
          CUD0SELECTSLU,
          DATAF,
          DATAFAM,
          DELAYBINDRESP,
          ECHODTEADDR,
          ECHOFAC,
          EMSGE,
          IDTST,
          IMS,
          INHIBITBIDREJ,
          LCN0USED,
          LLC0CTCPCHK,
          LLC5CTCPCHK,
          MCHTMR={seconds|60},
          NOCLOSEONTAPFAILURE,
          NORTRBIDREJ,
          NRITAB=name,
          ONEPIUINB,
          {PRI|SEC|PEER},
          PFXDCEADDR,
          PVCRECONTMR={seconds|60},
          PVCSETUPREJ=code,
          PVCSETUPTMR={seconds|60},
          REPDCEADDR,
          REQSESSDELAY={seconds|2},
          RESETINO,
          RESIDSTART={firstid|pvcct+1},
          RETPIU,
          REUSEBUSYSPU,
          STRIPFAC,
          STRIPRTEIN,
          SVCCALLTMR={seconds|30},
          TAPWITHCLR,
          TCPRBLMT={count|{7|2})
          XID={NO|STD|TAB|TABSTD|(idnum,idcnt)}]

                                        (DFX|SPU|MCH|XTP client)
                                              (new for V2R1M0)
                                          (changed for V2R1M1)
                                          (changed for V2R2M0)
                                          (changed for V2R3M0)
                                          (changed for V2R4M0)
```

Specifies special processing options for this REMOTE definition statement.

CLOTINITYP={<u>NONE</u>|BIND|TIMER|CONSOLE} for TYPE=SPU REMOTEs only, specifies the type of outbound call initiation that will be used for a QLLC SPU. TYPE=SPU must also be specified.

CLOTINITYP=<u>NONE</u> specifies that an outbound QLLC call *will not be* initiated for this SPU. **NONE** is the default value that will be used if the CLOTINITYP= option is not specified or if the specified value is invalid. See SVC3= operand description for an alternative method of QLLC callout initiation.

CLOTINITYP=BIND specifies that an outbound QLLC call *will be* initiated when any SLU defined in the LUNAME= operand for the SPU is bound (acquired).

In order to provide this capability, HNAS will open the ACBs and condition all SLUs to accept a BIND request. The first BIND directed at an SLU in the LUNAME= operand list will initiate the QLLC callout attempt for the SPU. All BINDs received by HNAS while the call establishment procedure is in progress will be queued for delivery to the target SLUs until the SPU and the SLU sessions come active. In other words, once the QLLC call process has been started, BINDs are held until the following sequence is performed:

1) the call is accepted,
2) an QXID request/response is exchanged to verify the IDBLK/IDNUM values,
3) a QSM request/QUA response is exchanged to activate the SPU QLLC component,
4) an ACTPU request/response is exchanged to activate the SPU SNA component,
5) ACTLU requests/responses are exchanged to active the SLUs SNA component.

Note: A BIND will remain queued even after the ACTLU response is received if the response carries the ‘power off’ indication in Control Vector 12. In this case, the BIND will remain queued until a NOTIFY request is received that carries the ‘power on’ indication. Once the ‘power on’ indication has been detected, the BIND will be sent to the target SLU. No LOGTAB or USSTAB processing is required, and hence, no USSMSG10 will be transmitted because the PLU is the initiator of the SLU connection.

CLOTINITYP=TIMER specifies that an outbound QLLC call *will be* initiated automatically when the OPTIONS=MCHTMR=*value* expires for the *first* defined MCH.

Contrast CLOTINITYP=TIMER to the timer initiated QLLC callout based on the SVC3= operand and MCHTMR= option value for a TYPE=MCH REMOTE definition statement. CLOTINITYP=TIMER support provides additional flexibility by allowing the number of callout attempts to be limited by the CLOTCONLMT= option. See QLLC callout summary below and the description of the SVC3= operand for additional information.

CLOTINITYP=CONSOLE specifies that an outbound QLLC call *will be* initiated when the VARY ACT console command is entered for the named SPU.

Note: When OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE is specified for a TYPE=SPU REMOTE definition statement, the DTEADDR=, DCEADDR=, FAC= and CUD= operands on the same TYPE=SPU REMOTE definition statement provide, respectively, the ***called*** DTE address, ***calling*** DTE address, facilities and call user data for the outbound Call

Request packet. For all CLOTINITYP= values, an established call remains active until cleared by the remote SPU or by HNAS if there is no VC activity for the period specified by the IDLETO= value.

Note: If an SPU is defined with a CLOTINITYP= value and it is also referenced in the SVC3= operand for a TYPE=MCH REMOTE definition statement as a callout SPU (the character O follows the associated DTE address), CLOTINITYP= call initiation will supersede SVC3= callout initiation. See QLLC callout summary below and the description of the SVC3= operand for additional information.

CLOTFAILRTYLMT=*count* for TYPE=SPU REMOTEs only, specifies a count value that is used to limit the number of callout retries in the event of a call failure. A default value of 3 is used if *count* is omitted. If a zero *count* is specified, failed calls will be retried indefinitely until a connection is established. TYPE=SPU and OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE must also be specified.

CLOTCONLMT=*count* for TYPE=SPU REMOTEs only, specifies a count value that is used to limit the number of callout attempts after a successful connection has been cleared. A default value of 0 is assumed if *count* is omitted. If a zero *count* is specified or is set by default, HNAS will attempt to reestablish the connection to the SPU indefinitely after a forced delay based on the CLOTINITYP= value. TYPE=SPU and OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE must also be specified.

This operand is provided to limit the use of X.25 network facilities for an SPU. When the number of call attempts reaches the CLOTCONLMT= *count* value, a subsequent outbound QLLC call can only be initiated via the VARY ACT console command.

Note: Although the CLOTCONLMT= option may be specified when a CLOTINITYP= value other than NONE is also specified, it is designed primarily to be used to control timer based callout initiation requests so that these automatic requests do monopolize X.25 network facilites. When the CLOTCONLMT= value is reached, subsequent outbound call attempts are blocked. If the CLOTCONLMT= option is specified for CLOTINITYP=BIND, BINDs will be rejected once the limit is reached. For this reason, we recommend omitting the CLOTCONLMT= operand when CLOTINITYP=BIND is specified. This will allow BINDs to always initiate a QLLC callout attempt. The CLOTCONLMT= operand has no meaning for CLOTINITYP=CONSOLE since this is the only way to restart a QLLC callout attempt that has been blocked when the CLOTCONLMT= value is reached.

Note: The CLOTCONLMT= option is only used when an outbound QLLC call is initiated based on the CLOTINITYP= option for an SPU. It is not used to control timer based callout initiation requests that result from the SVC3= operand and MCHTMR= value on a TYPE=MCH REMOTE definition statement.

QLLC Callout Summary:

QLLC Callout via Application Bind

HNAS will initiate a QLLC SVC connection for an SPU when an Application Bind is received on *any* one of the SLUs identified in the SPU's LUNAME= operand if, and only if, OPTIONS=CLOTINITYP=BIND is specified for the SPU. The Call Request packet will carry information provided by the CUD=, DCEADDR=, DTEADDR= and FAC= operands for the SPU.

PLU BINDs that are received before the SPU is connected are queued for delivery after the SPU and its SLUs come active. After the SPU connection is established, an XID request/response exchange is performed followed by an ACTPU and ACTLUs for all defined SLUs in the LUNAME= operand. Any BINDs that have been queued for transfer are then passed to the target SLUs. In the case of a queued BIND, no SLU input and hence no REQSESS is required to establish a SLU/PLU session. The queued BIND already does this. If the SPU is already connected when a BIND is received, the BIND is treated in the normal fashion and is immediately passed to the remote SLU.

<u>QLLC Callout via Timer Control</u>

HNAS can initiate a QLLC SVC connection based on a timer. There are two (2) methods for timer initiated callout.

1) If OPTIONS=CLOTINITYP=TIMER is specified for an SPU, a callout is initiated when the MCHTMR= value expires for the *first* MCH defined in the CDF. This MCH does not have to be associated with any SPU. The MCHTMR= timeout is used solely to provide an event that initiates SPU callout processing. The Call Request packet will carry information provided by the CUD=, DCEADDR=, DTEADDR= and FAC= operands for the SPU rather than the MCH. The DTEADDR= operand for the SPU supplies the ***called*** DTE address.

2) If an SPU is identified in the SVC3= operand for an MCH and the SVC3= entry has a ***called*** DTE address (e.g., SVC3=(,...,*spuname*/*dteaddr*O,...)), HNAS will initiate a connection for the SPU when the MCHTMR= value expires for *this* MCH. The Call Request packet will carry information provided by the CUD=, DCEADDR= and FAC= operands and the SVC3= ***called*** DTE address for the MCH rather than the SPU.

Note: OPTIONS=CLOTINITYP=TIMER callout initiation takes precedence over SVC3= callout initiation. That is, HNAS will look for all SPU with OPTIONS=CLOTINITYP=TIMER before it looks for any identified in an SVC3= operand.

<u>QLLC Callout via HNAS Console Control</u>

HNAS will initiate an QLLC SVC connection for an SPU when the following HNAS Console Command is executed if OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE is specified:

RNM=*spuname* VARY ACT

In the case of OPTIONS=CLOTINITYP=CONSOLE, this command is the *only* way to initiate an SPU callout.

In the case of OPTIONS=CLOTINITYP=BIND|TIMER, this command is the *only* way to initiate an SPU callout after the call connection limit (OPTIONS=CLOTCONLMT=*count*) has been reached. The Call Request packet will carry information provided by the CUD=, DCEADDR=, DTEADDR= and FAC= operands for the SPU.

QLLC Callin/Callout Contention

If an inbound QLLC call targets an SPU which is identified for callout usage via the CLOTINITYP= option, the inbound call will be accepted and the SPU will be allocated if the SPU is not currently connected, an outbound call attempt is not in progress and the IDBLK/IDNUM values for the inbound call matches the values configured for the SPU. This allows the same SPU to be used for both inbound and outbound calls.

CUD0SELECTSLU for a TYPE=MCH|MXT REMOTE specifies that the CUD0 byte from an inbound Call Request packet will participate in the selection of an LLC0 or LLC5 LU for an inbound call. When a ‘X’ entry in an SVC0/5 TYPE=MCH REMOTE is processed HNAS selects the LU based on a compare of the hex digits in an ‘X’ entry and call request packet data starting at COD0 (option specified) or CUD1 (option not specified). The OPTION provides capabilities that are found in IBM Spain’s ISARDX25 37xx product.

Note: The CUD0SELECTSLU option was added as part of Enhancement APAR 2400074.

DATAF|DATAFAM (for Datafono support)

Select one of the two values shown above to specify the processing for Datafono terminal responses to type ‘M’ messages from the PLU. For additional information on these OPTIONS= values, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

DATAF for TYPE=DFX REMOTEs only, specifies that when an ‘M’ message is sent to the remote HNAS delivers the DR+ (if requested) to the PLU as soon as the ‘M’ message is transmitted. When the D(9) response is received from the remote it is sent to the PLU as an input message.

DATAFAM for TYPE=DFX REMOTEs only, specifies that when an ‘M’ message is sent to a remote HNAS waits for the D(9) response from the remote. When the D(9) response arrives it is discarded and a DR+ response (if requested) is sent to the PLU. The PLU does not need logic for the D(9) message.

Note: The NATIVE and NATIVENV device types supported by XAI.ZSWTC define LLC0 resources. HNAS does not support these types in the OPTIONS= list. Most ISARD LLC0 support options are available in the base HNAS product. Changes may be required because functions provided by NRI tables are provided by SVC0= operands on HNAS TYPE=MCH REMOTE statements. Pseudo leased-support is not provided for HNAS LLC0 devices.

DELAYBINDRESP for TYPE=MCH|XTP REMOTEs, specifies that HNAS is to delay the response to a BIND that triggers an LLC0 or LLC5 callout operation until the call succeeds (call accept received, +RSP) or fails (clear or is timeout, -RSP). The sense data for a -RSP will be 0801C3D9 - resource not available.

ECHODTEADDR for TYPE=MCH REMOTEs only, specifies that the DTE address field (both the ***called*** and ***calling*** DTE addresses) from the **inbound** Call Request packet is to be echoed back to the remote DTE in the Call Accept packet that HNAS generates for non-GATE XOT connections. TYPE=MCH must also be specified.

Note: For GATE connections, the Call Accept packet comes from the CTCP which may or may not echo the DTE address field.

ECHOFAC for TYPE=MCH REMOTEs only, specifies that the facilities field from the **inbound** Call Request packet is to be echoed back to the remote DTE in the Call Accept packet that HNAS generates for non-GATE XOT connections. Some networks do not support this feature while others can use the response to further negotiate facility values. TYPE=MCH must also be specified.

Note: For GATE connections, the Call Accept packet comes from the CTCP which may or may not echo the facilities field.

EMSGE for TYPE=DFX REMOTEs only (for Datafono support), specifies that HNAS is to send a ‘ESPERE POR FAVOR’ ‘M’ message if the PLU does not deliver a response in 26 seconds. If this option is not coded, HNAS will not generate the message. For additional information on this OPTIONS= value, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

IDTST for TYPE=DFX REMOTEs (Datafono support), specifies that the IDTST ISARDX25 option applies to the session. With this OPTION the R(3) initialization message from the remote device provides a 6 or 11 character session ID for the call. The ID is bounded by X’23’ and X’2D’ characters. HNAS records the ID string and delivers the R(3) message to the PLU. When the PLU sends a message in an IDTST session, the message must start with an 11 character field containing the PLU’s ID string. If the PLU’s ID string does not match the HNAS ID string then a NAS5725W LU lu-name DISCARDING MSG W BAD ID FROM PLU plu-name alert is issued and the PLU’s message is discarded. This option is designed to prevent messages queued in the PLU from being delivered to the wrong remote.

Note: The IDTST option was added as part of Enhancement APAR 2400074.

IMS for TYPE=DFX REMOTEs only (for Datafono support), specifies that the PLU is. With this parameter, instead of a DR-, HNAS will send a DR+ (if solicited) and will ignore the application response in the following cases:

1) When receiving ‘M’ messages from the PLU and the RETPIU option is set (see below).

2) When a message of incorrect type is sent to HNAS by the PLU. A message that does not start with ‘M’, ‘R’, ‘D’ or ‘L’ has an incorrect type.

3) When a message is received outside of the normal sequence of the Datafono protocol.

If the option is not coded, the above actions are not taken.

For additional information on this OPTIONS= value, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

INHIBITBIDREJ for TYPE=MCH|XTP REMOTEs, specifies that the HNAS will not reject, but will always accept a BID request from the host PLU even if the HNAS SLU has input data pending and it was bound as the first speaker. Normally, in this case, the HNAS SLU would reject the BID with 0814 or 0813 sense (RTR is or is not forthcoming). The 0814 sense is used if the SLU is not allowed to end a bracket (an RTR is sent to the PLU after the SLU’s input has been completely processed). The 0813 sense is used if the SLU is allowed to end a bracket (the SLU will signal end bracket after all of it’s input has been processed).

Some host applications cannot handle either BID reject condition. These applications will UNBIND the SLU if a BID is ever rejected. For these applications, INHIBITBIDREJ should be specified. In this case, HNAS will hold any pending input until the PLU signals that it is ready to accept the input using the end bracket indicator (EBI) or change direction indicator (CDI). TYPE=XTP|MCH must also be specified.

Note: This problem was first noticed with CICS. It seems to be related to the way CICS programs (transaction processors) interface to CICS. Unfortunately, the only way to tell that the INHIBITBIDREJ option is needed is to observe the failure. Regrettably, this usually requires a trace. The defaults are correct for the majority of installations. For V2R4M0, a NAS3705W alert message is generated when HNAS rejects a PIU from the host. If the sense data in the message is 0813 or 0814 then use of this option may reduce or eliminate the alerts.

LCN0USED for TYPE=MCH|XTP REMOTEs, specifies that LCN0 is to be included in the calculation of the GATE Resource Identifier (RESID) that HNAS assigns to all GATE terminal session SLUs. Each terminal session SLU in the SVC4= operand (as well as the LLC4 SLU entries in the PVC= operand) is assigned a unique RESID that identifies the terminal session SLU to the CTCP over the control session SLU connection. Normally, the RESID starts at one (1) plus the number of PVCs configured (PVC= operand ***vclmt*** value). LCN0USED tells HNAS to start the RESID at zero (0). This is required for some CTCPs (CSFI for example) that know and use the NPSI LCN0=USED operand. TYPE=MCH|XTP, GATE=GENERAL and PVC=NONE must also be specified.

LLC0CTCPCHK and/or **LLC5CTCPCHK** for TYPE=MCH REMOTEs only, specify that HNAS is to examine the first GATE CTCP SLU in the LUNAME= operand list when an LLC0 (PCNE) and/or LLC5 (PAD) call, respectively, is received. If this CTCP SLU is not bound, the new call is cleared with a diagnostic code of 254 (X'FE'). If the CTCP SLU is bound, call request processing is performed in the normal manner. This option was added so that an X.25 service provided like Transpac can reroute a call that has been clear with the 254 diagnostic code. TYPE=MCH, GATE=GENERAL and CONNECT=NO must also be specified.

MCHTMR={*seconds*|60} for TYPE=MCH|XTP REMOTEs, specifies that HNAS is to interrogate the GATE SLUs associated with this MCH at ***seconds*** intervals to ensure that those that are idle can be made active in order to reduce or eliminate VC clears with DIAG=X'85' ('GATE: selected CTCP not active'). TYPE=XTP|MCH must also be specified.

During MCH timer service, GATE control session SLUs are continually interrogated to ensure that they are in the active state (BIND/SDT) to accept incoming calls. Normally, GATE SLUs are checked at one minute intervals. In some instances, this may not be often enough to ensure that their activation occurs before a Call Request packet is received and thus prevent a clear with DIAG=X'85'.

The **MCHTMR=** option allows you to specify an interval less than one minute which instructs HNAS to monitor the GATE SLUs more often than it normally would. Sub-minute monitoring can reduce or eliminate DIAG=X'85' clear conditions.

Note: The **MCHTMR=** option also triggers HNAS to query all terminal SLUs to ensure that their Application Control Blocks (ACBs) are OPEN so that they can be acquired (accept BIND requests) for callout. Each ACB that HNAS OPENs requires a corresponding APPL statement in the HNAS Application Major Node File (AMNF).

You may specify a ***seconds*** value between 4 and 60. If the **MCHTMR=** option is not specified or if the specified ***seconds*** value is invalid, a default value of **60** (seconds) will be used.

For example, OPTIONS=MCHTMR=16 will cause HNAS to check the GATE SLUs every 16 seconds.

NOCLOSEONTAPFAILURE for TYPE=XOT|XTP REMOTEs, specifies that the HNAS will not close active sockets and their VC/LU components when a router contact loss condition is detected, that is, when 2 consecutive TAP (keep alive) failures are detected.

Currently, if 2 consecutive TAP failures occur, HNAS assumes that the remote router is down and then resets all the active TCPIP sockets associated with the failed router. All LU/VC connections to the router are also closed.

Sometimes, a customer would like to monitor the TAP process (MONTAP) and leave the connections active even when it appears that the router is down.

HNAS has been modified to accept NOCLOSEONTAPFAILURE as a REMOTE option. When OPTIONS=NOCLOSEONTAPFAILURE is specified for a TYPE=XOT|XTP REMOTE, a TAP failure will not cause all active connections to the router to be closed. The sessions will remain active even though the router is marked as offline. When a TAP failure occurs, the following message is generated based on the setting of the {NO}CLOSEONTAPFAILURE option.

```
NAS2502E CLIENT=010.117.056.100(02704) SOCKID=0001 PCEID=000B NAME=R1CNIN
NAS2502E ROUTER CONTACT LOST, CLOSEONTAPFAILURE OPTION IS IN EFFECT
```

```
NAS2505E CLIENT=010.117.056.100(02704) SOCKID=0001 PCEID=000B NAME=R1CNIN
NAS2505E ROUTER CONTACT LOST, NOCLOSEONTAPFAILURE OPTION IS IN EFFECT
```

In addition, this option can be toggled using the MRMT command as follows:

MRMT *rmtname* OPTIONS={NO}CLOSEONTAPFAILURE

The DRMT command will display the current TAP failure option. CLOSEONTAPFAILURE can only be specified as an MRMT option and is the default when NOCLOSEONTAPFAILURE is not specified in the CDF.

The MONTAP NAS2511M message was also modified to display a retry count (*ddddd*). The count is reset when a successful TAP response is received. The count indicates the number of unsuccessful TAP attempts since the last successful TAP exchange.

```
NAS2511M CLIENT=010.117.056.100(02704) SOCKID=0001 PCEID=000B NAME=R1CNIN
NAS2511M XOT TAP TIMEOUT, RESPONSE NOT RECEIVED FOR CONNECTION SETUP (ddddd)
```

Note: If MRMT *rmtname* OPTIONS=CLOSEONTAPFAILURE is entered when the NOCLOSEONTAPFAILURE is in effect and router contact was lost (NAS2505E message was issued), all active connections to the down router are immediately closed and the following message is generated:

```
NAS2507E CLIENT=010.117.056.100(02704) SOCKID=0001 PCEID=000B NAME=R1CNIN
NAS2507E ROUTER CONTACT LOST, CLOSEONTAPFAILURE OPTION IS IN EFFECT
```

This message is identical to the NAS2502E message except for the message ID.

Note: The NOCLOSEONTAPFAILURE option was added as part of Enhancement APAR 2400055.

NORTRBIDREJ for TYPE=MCH|XTP REMOTEs, specifies that the HNAS will not reject a BID request from the host PLU with 0814 sense (RRT is forthcoming) even if the HNAS SLU has input data pending, was bound as the first speaker and is not allowed to end a bracket. Normally, in this case, the HNAS SLU would reject the BID with 0814 sense (see INHIBITBIDREJ for additional information).

Some host applications cannot handle the 0814 BID reject condition. These applications will UNBIND the SLU if a BID is rejected with 0814 sense. For these applications, NORTRBIDREJ should be specified. In this case, HNAS will only reject the BID with 0813 sense (RTR is not forthcoming). After all SLU input data is passed to the PLU, the SLU will wait for the PLU to either end the bracket (EBI) or give the SLU permission to send another block of data (CDI). If EBI is given, the dialog reverts to contention state in which case either the SLU or PLU can start another bracket. TYPE=XTP|MCH must also be specified.

Note: This problem was first noticed with CICS. It seems to be related to the way CICS programs (transaction processors) interface to CICS. Unfortunately, the only way to tell that the NORTRBIDREJ option is needed is to observe the failure. Regrettably, this usually requires a trace. The defaults are correct for the majority of installations. For V2R4M0, a NAS3705W alert message is generated when HNAS rejects a PIU from the host. If the sense data in the message is 0814 then use of this option may reduce or eliminate the alerts.

NRITAB=*name* for TYPE=DFX REMOTEs only (for Datafono support), specifies the name of an NRI table used to convert a calling DTE address to an IDNUM value. HNAS loads the table from a library identified by the VTAMLIB DD statement. OPTIONS=XID={TAB|TABSTD} must also be coded. For additional information on this OPTIONS= value, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

ONEPIUINB for TYPE=MCH|XP REMOTEs, specifies that the HNAS will not send more than one RU chain to the PLU when input data is pending, bracket state is active and the SLU owns the change direction indicator (the PLU has given permission for the SLU to send data). Normally, in this case, the HNAS SLU would send another RU chain to the PLU and give up the change direction indicator after all of the input data had been processed (see INHIBITBIDREJ and NORTRBIDREJ for additional information).

Some host applications cannot handle more than one inbound RU chain within a bracket. These applications will UNBIND the SLU if multiple inbound RU chains are received within a bracket. For these applications, ONEPIUINB should be specified. In this case, HNAS will hold any pending input if bracket state is active (due to previous SLU input) until the PLU BIDs for the change direction indicator (CDI), sends its data and then signals that it is ready to accept additional SLU input using the end bracket indicator (EBI). TYPE=XTP|MCH must also be specified.

Note: When the ONEPIUINB option is in effect, only one (1) complete RU chain is sent to the PLU for one complete Mbit (M-bit) packet chain.

Note: To demonstrate the reason for the ONEPIUINB option, assume MBITCHN=YES is in affect and brackets are in use. HNAS will deliver an X.25 Mbit packet chain as an RU chain using the RU size carried in the BIND. After the RU chain has been delivered and a bracket started (BBI), if a second X.25 Mbit packet chain arrives, there are two ways to handle it:

1) Send the second X.25 Mbit packet chain as a new RU chain to the PLU with BBI off (INB state). This works for some and applications but not for all.

2) Wait for the PLU to end the current bracket (EBI) then deliver the second X.25 Mbit packet chain as a new RU chain with BBI set.

HNAS and our earlier 37xx NAS product, by default, use scenario #1 above. The ONEPIUING option forces scenario #2 above. This problem was discovered in Spain using CICS. For both scenario #1 and #2, the processing applies to all LLCs.

Caution: While the ONEPIUINB option may or may not be required for some CICS systems, it should not be used for TSO. We have observed that line mode TSO sessions do not work when the ONEPIUINB option is in affect.

{PRI|SEC|PEER} for TYPE=SPU REMOTEs only, specifies whether this SPU takes the roll of the primary, secondary or peer link station. Primary link stations can send QLLC requests only. Secondary link stations can send QLLC responses only. Peer link stations can send QLLC requests and responses. If you are not sure of the type of link station at the remote end, specify **PEER** or allow the type to default. TYPE=SPU must also be specified.

PFXDCEADDR for TYPE=MCH REMOTEs only, specifies that the value coded for the **DCEADDR=** operand is to prefix (appended in front of) the ***calling*** DTE address in a GATE **outbound** Call Request packet when a ***calling*** DTE address (or subaddress) is provided by the CTCP. If no ***calling*** DTE address is supplied by the CTCP (field is null), PFXDCEADDR is treated as REPDCEADDR. TYPE=MCH and GATE=GENERAL must also be specified.

PVCRECONTMR=*seconds* for TYPE=MCH|MXT REMOTEs, specifies how often HNAS tries to re-establish a session between a PVC and its PLU. The VTAM session with the PLU is established after the PVC session between HNAS and the router is established by PVC SETUP packets. If the PVC's VTAM session is ended (via PLU UNBIND, NOTIFY or TPEND) then the value specified by this operand controls how often HNAS will try to reconnect the session with the PLU.

You may specify a ***seconds*** value between 5 and 254. If PVCRECONTMR= is omitted for a TYPE=MCH REMOTE, a default value of 60 seconds will be used. If PVCRECONTMR= is omitted for a TYPE=MXT REMOTE that is associated with a PVC, no default value is set so that the root MCH value will be used.

PVCSETUPREJ=*code* for TYPE=MCH|MXT REMOTEs, specifies a PVC SETUP reject status code that will be used any time HNAS receives a SETUP for the PVC. Since a status code greater than X'0F' causes the router to stop sending SETUP packets for the PVC, this option may be used to shut down PVC SETUP initiation by the router.

You may specify a hexadecimal ***code*** value (without the framing characters X'') between 01 and 7F except for 12. If PVCSETUPREJ= is omitted, SETUP packets are processed normally (status code to connect or reject the session generated by HNAS). This parameter was initially implemented for PVC setup testing but also includes benefits as described above.

PVCSETUPTMR=*seconds* for TYPE=MCH|MXT REMOTEs, specifies how often HNAS should send PVC SETUP packets to establish a PVC session between HNAS and the router. This option only has meaning when the PVC definition in the HNAS CDF identifies the router to be used for the session (see PVC= operand).

You may specify a ***seconds*** value between 10 and 254. If PVCSETUPTMR= is omitted for a TYPE=MCH REMOTE, a default value of 60 seconds will be used. If PVCSETUPTMR= is omitted for a TYPE=MXT REMOTE that is associated with a PVC, no default value is set so that the root MCH value will be used.

REPDCEADDR for TYPE=MCH REMOTEs only, specifies that the value coded for the **DCEADDR=** operand is to be placed in the ***calling*** DTE address field in a GATE **outbound** Call Request packet when no ***calling*** DTE address is provided by the CTCP (field is null). TYPE=MCH and GATE=GENERAL must also be specified.

REQSESSDELAY=*seconds* for TYPE=MCH|XTP REMOTEs, specifies the amount of delay time that HNAS will use between the delivery of a GATE Call Request packet to the CTCP control session SLU and the a VTAM REQSESS which solicits a BIND from the PLU to start the GATE data session SLU. In the past, no delay was enforced which, in some cases, caused the REQSESS to be rejected. The default (2) is correct for most installations. Some file transfer applications (e.g. CFT) may require the delay. When the delay is required, the GATE data session SLU does not start (no BIND from the PLU in response to the HNAS REQSESS). Some GATE callout applications don't require the delay (2 second default) so the delay can be eliminated by coding a value of 0.

You may code a ***seconds*** value between 0 and 254. If omitted, 2 seconds will be used. TYPE=XTP|MCH and GATE=GENERAL must also be specified.

RESETINO for TYPE=MCH REMOTEs, specifies that when a RESET is received HNAS is to end the PLU session with a NOTIFY (generated when the LU's HNAS ACB is closed). The NAS3799I session end message will show "HNAS CAUSE/DIAG=000/211". The X25 session that initiated the RESET is also ended with a CLEAR CAUSE/DIAG=000/211.

When RESETINO is in effect:

For LLC0 and LLC5 SVCs:
Any inbound RESET terminates the PLU session and ends the X25 session with CLEAR C/D=000/211.

For LLC0 and LLC5 PVCs:
Inbound RESETs with C/D=000/015 or 000/000 are ignored (these are used by the router to report PVC operational status). Any other inbound RESET terminates the PLU session with a NOTIFY. The X25 PVC session remains active.

For LLC3 sessions:
Because an inbound RESET can potentially affect many PLU sessions HNAS has always terminated all associated PLU sessions when a reset is received (i.e. this option has no effect. As above, C/D=000/015 and 000/015 RESETs are excepted for PVC QLLC sessions.

LLC4 sessions:
An inbound RESET is passed to the GATE PLU where the processing decision is made.

Note: The RESETINO option was added as part of Enhancement APAR 2400081.

RESIDSTART={*firstid*|`pvcct+1`} for TYPE=MCH|XTP REMOTEs, specifies the first resource identifier value to be used for **inbound** Call Request packets that are passed to the GATE CTCP. If omitted, the first resource identifier is generated based on the LCN0USED option and the PVC= operand count. TYPE=XTP|MCH and GATE=GENERAL must also be specified.

You may specify a ***firstid*** value between 0 and 4095. If the **RESIDSTART=** option is not specified or if the specified ***firstid*** value is invalid, a default value of the PVC count (***vclmt***) plus one (1) will be used.

RETPIU for TYPE=DFX REMOTEs only (for Datafono support), controls the actions to be taken when a dataphone terminal session ends in an unexpected manner, i.e., without the transmission of an 'L' message. When there is an unexpected termination (e.g. Clear packet received), HNAS would normally send a -RSP to the PLU and close the HNAS VTAM ACB for the session. This causes a NOTIFY PIU to be delivered to the PLU. The recovery mechanisms of the IMS will make the PLU indefinitely attempt the sending of queued responses to the terminal. When the RETPIU and IMS options are coded the following steps are taken when there is an unexpected clear:

1) The HNAS LU resource is marked ‘busy’ so it is not available for another call.

2) HNAS sends +RSP to messages from IMS and discards the messages until a non-’M’ message is received from the PLU.

3) The first message that was not sent to the remote is sent to the PLU as input. The message will be prefixed by a ‘?’ character

If the option is not coded, the above actions are not taken.

For additional information on this OPTIONS= value, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

REUSEBUSYSPU for TYPE=SPU REMOTEs only, specifies, for a TYPE=SPU REMOTE, that HNAS is to allow the SPU to be allocated to a new VC call even when it is already connected to a VC. TYPE=SPU must also be specified.

This option may be required if the QLLC PAD fails to notify HNAS of an SPU disconnect. When the REUSEBUSYSPU option is effect, the old VC call is cleared and the new VC call is allowed to proceed. When the REUSEBUSYSPU option is *not* in effect, the new VC call is cleared and the old VC call is allowed to continue. In the latter case, the following alarm message is generated:

```
NAS8102W PU FOR DTEADDR=DDDDDDDDDDDDDDD IDBLK/IDNUM=xxxxxxxx
                ON MCH mchname
NAS8102W WAS IN USE WHEN SELECTED
```

STRIPFAC for TYPE=MCH REMOTEs only, specifies that all facilities data is to be removed from an **inbound** Call Request packet before the packet is passed to the GATE CTCP. TYPE=MCH and GATE=GENERAL must also be specified.

STRIPRTEIN for TYPE=MCH REMOTEs only, specifies that the ***called*** DTE address from an **inbound** Call Request packet is to be removed before the packet is passed to the GATE CTCP. TYPE=MCH and GATE=GENERAL must also be specified.

Only that portion of the ***called*** DTE address that matches the associated **RTEIN** operand value is removed. For example, if RTEIN=MCH1/20361005 is specified for the TYPE=XOT LOCAL definition statement and STRIPRTEIN is in affect for MCH1, only the digits following 20361005 will remain in the ***called*** DTE addresses field of the inbound Call Request packets that are passed to the GATE CTCP.

If the RTEIN= entry used to route the call uses the calling address (e.g. RTEIN=(1234S/MCH1) then the called dte address in the call request packet is passed to the CTCP without modification.

If the RTEIN= entry used to route the call uses the 'match at end option' (e.g. @1234/MCH1) then the leftmost digits before the rightmost matching digits are removed from the called address passed to the CTCP. For example, if the called address is 76761234 and RTEIN=(@1234) then 1234 will be passed to the CTCP as the called address.

The STRIPRTEIN option is processed only after subaddress LLC selection is performed (SUBADDR=YES and LLC*i*=(...)) and only when the established LLC is LLC4 (GATE).

SVCCALLTMR=*seconds* for TYPE=MCH|MXT REMOTEs, specifies how long HNAS should wait for a Call Accepted or Clear Request response to an outbound Call Request packet. When the timer expires HNAS assumes that the call has failed and steps to the next RTE-OUT= entry that can be used for the call. The timeout indicates that either something was wrong with the Call Request packet or some type of network problem has occurred.

You may specify a ***seconds*** value between 10 and 254. If SVCCALLTMR= is omitted for a TYPE=MCH REMOTE, a default value of 30 seconds will be used. If SVCCALLTMR= is omitted for a TYPE=MXT REMOTE that is associated with a SVC, no default value is set so that the root MCH value will be used. The value coded on a TYPE=MXT REMOTE is only used for LLC0 and LLC5 calls that reference the MXT (see SVC0= and SVC5= callout descriptions). LLC3 and LLC4 calls do not use MXTs so the timeout value comes from the MCH SVCCALLTMR= option.

Note: T21= can be specified as an alternate name for the SVCCALLTMR= suboperand.

Note: OPTIONS=SVCCALLTMR= support was introduced into 240 as Enhancement APAR 2400069.

TAPWITHCLR for TYPE=XOT REMOTEs only, specifies that HNAS Keep Alive logic (**TAP=nn**) should use a Clear Request rather than a Call Request as the shoulder TAP request. TYPE=XOT must also be specified.

Note: For HNAS release V2R2M0, Keep Alive logic only used a Call Request packet as the shoulder TAP request packet. Some router configurations cause 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 prior to 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. The TAP request was changed to a Call Request for V2R2M0 to ensure that a response from the router is returned 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.

TCPRBLMT={*count*{7|2} for TYPE=XOT|XTP REMOTEs, specifies the number of staging buffers that HNAS is to preallocate for each TCPIP socket to handle inbound data transfers. These buffers are allocated from the HNAS system buffer pool which is created using the BFRLMT= and BFRSIZ= operand values on the BUILD definition statement. TYPE=XTP|XOT must also be specified.

In releases prior to V2R2M0, the number of pre-allocated staging buffers was maintained as an internal value of seven (7). For XTP sockets, this was a reasonable number because all VC connections are multiplexed across a single TCPIP socket. For XOT VCs, however, this value can be excessive because each VC uses a separate TCPIP socket. It has been observed that for XOT VCs, a value of two (2) is adequate for most transactions. Configuration environments with hundreds of XOT VC connections will benefit from the reduced memory requirement that results from using a small staging buffer count.

Note: If you plan to support QLLC (LLC3) resources, we recommend coding TCPRBLMT=7 instead of allowing the default of 2 to be used for XOT VCs. This is because the data traffic for all QLLC SLUs on an SPU is multiplexed across a single QLLC VC socket in much the same way as all non-QLLC VCs are multiplexed across a single XTP socket.

You may specify a ***count*** value between 1 and 7. If the **TCPRBLMT=** option is not specified or if the specified ***count*** value is invalid, a default value of **7** or **2** will be used for XTP or XOT sockets, respectively.

XID={NO|STD|TAB|TABSTD|(*idnum,idcnt*)} for TYPE=DFX REMOTEs only (for Datafono support), specifies how HNAS is to locate an SLU resource for the Datafono session. The SLU will look like the VIRTUAL=YES resources used in a NPSI environment. XID=STD is assumed if XID= is omitted.

NO specifies that there is no XID for the session. HNAS locates an available pseudo-leased LU from one of the TYPE=DFL REMOTEs addressed by the DFLNAME= parameter on the TYPE=MCH REMOTE with the call request. A pseudo-leased resource is available if the HNAS SLU has no session with a remote and if the LU is active (BIND, SDT received from the PLU).

STD specifies that an available LU for the session is to be located by searching the switched LU pool for a resource with an IDNUM value matching the IDNUM value in call user data bytes 1 through 3. IDNUMs are 5 (hex) digit values. The 6 digit value taken from CUD1-3 is right shifted four bits to obtain the search value. When an appropriate resource is located HNAS opens the HNAS ACB for the resource and requests a session with the PLU associated with the resource. The switched LU pool is created by TYPE=DFS REMOTE statements.

TAB specifies that an IDNUM value is to be generated by searching the NRI table addressed by NRITAB= for an entry with a DTE address matching the calling DTE address in the call request packet. The call is cleared if no entry is found.

TABSTD is treated the same as **TAB** except that if the IDNUM value located in the NRI table is zero then act as though STD were coded (generate IDNUM from CUD1-3).

(***idnum,idcnt***) specifies that HNAS is to search the switched LU pool for an available LU with an IDNUM in the range *idnum* to *idnum+idcnt*. *idnum* is a 5 digit hexadecimal number and *idcnt* is a decimal number in the range of 0 to 25.

For additional information on this OPTIONS= value, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

[PACE={{*pktcnt*|0}|(*rcvpacnt,sndpacnt*)}]

(SPU|XTP client)
(new for V1R1M2)
(changed for V2R2M0)

For a TYPE=XTP REMOTE definition statement, the PACE= operand specifies an output pacing count that represents the maximum number of non-Qualified Data packets that HNAS can send before it must exchange a pacing request/response.

PAD=INTEG|TRANSP|PACEONLY must also be specified because output pacing requires the use of Qualified Data packets for PAD communication.

Qualified Data packets are treated the same as non-Qualified Data as far as sequencing and flow control are concerned which means that the PAD must process them sequentially. When pacing is in effect, HNAS will send the number of packets specified by the ***pktcnt*** value followed by a PAD Set and Read Parameters Request to solicit a PAD Parameter Indication response. This serves as the pacing exchange and tells HNAS that the PAD has transmitted all previous non-Qualified Data packets. It further indicates that the router's transmit queue is empty and can now accept more data from HNAS. This exchange is required to prevent HNAS output from overrunning the router's transmit queue.

You may specify a ***pktcnt*** value between 0 and 127. If a ***pktcnt*** value is not specified or if the specified value is invalid, a default value of **0** will be used. A value of zero (0) indicates that pacing is suppressed.

For a TYPE=SPU REMOTE definition statement, the PACE= operand specifies global receive and sending pacing values for all the SLUs on this SPU.

The global receive pacing value (***rcvpacnt***) is used as the default SLU receive pacing value when the receive pacing value in the BIND image is zero (0) and a ***rcvpacnt*** value is not specified for the LUNAME= operand entry. You may specify ***rcvpacnt*** value between 0 and 253. If 0 is specified for ***rcvpacnt***, no receive pacing will be provided. If ***rcvpacnt*** is not specified or if the specified value in invalid, no receive pacing will be provided.

The global send pacing value (***sndpacnt***) is used as the default SLU send pacing value when the send pacing value in the BIND image is zero (0) and a ***sndpacnt*** value is not specified for the LUNAME= operand entry. You may specify ***sndpacnt*** value between 0 and 253. If 0 is specified for ***sndpacnt***, no send pacing will be provided. If ***sndpacnt*** is not specified or if the specified value in invalid, no send receive pacing will be provided.

[PAD={NO|INTEG|TRANSP|PACEONLY}]

(MCH|XTP client)

Specifies the Packet Assembler/Disassembler (PAD) functions that HNAS is to provide.

NO specifies that HNAS will not provide integrated or transparent PAD functions.

INTEG specifies that HNAS will take the role of integrated PAD and will handle all Qualified data exchanges.

TRANSP specifies that HNAS will take the role of transparent PAD so that a host application can handle all Qualified data exchanges. In this case, the first byte of all PIUs passed between HNAS and host application contains a data type value. The data type byte indicates whether subsequent data is Qualified or non-Qualified.

PACEONLY specifies that HNAS will not provide PAD services but will use Qualified Data packets to enforce output pacing if a PACE operand value is specified.

If the PAD operand is not specified or if the specified value is invalid, a default value of **NO** will be used.

```
[PADPARM={NONE|(pnum/pval,...)|
          1/0,7/21,8/0|
          1/0,7/2,8/0|
          3/2,4/0,7/2,13/4}]
```

(MCH|MXT|XTP client)

Specifies **NONE** or a list of X.29 PAD parameter numbers and values that will be used to set remote ITI PAD parameters when HNAS has an LLC5 session.

NONE specified as the first PADPARM= operand entry, indicates the no PAD parameters will be sent to the remote DTE at session establishment time.

You may specify a ***pnum*** value between 0 and 127. If a ***pnum*** value is not specified or if the specified value is invalid, the parameter number/value pair is ignored.

You may specify a ***pval*** value between 0 and 255. If a ***pval*** value is not specified or if the specified value is invalid, the parameter number/value pair is ignored.

In the past, HNAS sent the PAD parameters to the remote using a ‘Q’ packet containing a SET and READ parameters command. For V2R4M0, PAD parameters are set with a SET parameters command. The Q packet is sent (unless PADPARM=NONE is coded) just before the session with the PLU is started. If MCHSOL is being used (USS Table processing) the parameters are also sent before USS commands are solicited.

Example: `PADPARM=(7/4,13/0,3/4)`

HNAS provides the ability to send pad parameters for LLC5 transparent PAD (XPAD) sessions, a feature not available with NPSI. If you do not want parameters sent by HNAS in an XPAD session then omit the PADPARM= parameter and no defaults will be sent.

The following **default** PAD parameter values are assumed when the PADPARM=operand is not specified for PAD=INTEG resources:

PADPARM=(1/0,7/21,8/0)	for PAD=INTEG XOT sessions (NPSI defaults for XOT)
PADPARM=(1/0,7/2,8/0)	for PAD=INTEG XTP sessions (NPSI defaults for XTP)
PADPARM=(3/2,4/0,7/2,13/4)	for remote console sessions

For more information on X.29 PAD parameters, see Appendix A of this document.

If a PADPARM= operand value is specified for a referenced TYPE=MXT REMOTE definition statement, it always take precedence over the PADPARM= operand value from the root TYPE=MCH|XTP REMOTE definition statement.

[PKTSIZ={*bytecnt*|<u>256</u>}]

(XOT client only)
(restored for V2R2M0)

Specifies the default packet size for GATE callout sessions using this XOT router.

Specify a ***bytecnt*** value between 64 and 8192. If a ***bytecnt*** value is not specified or if the specified value is invalid, a default value of **256** (bytes) will be used.

Notes:

For XOT sessions established by the router the packet and window sizes are set as follows:

SVCs: Facilities field in the Call Request packet received by HNAS. A call request without packet and window size facilities is rejected with diag=128.

PVCs: Packet and window sizes are provided by the PVC Setup received by HNAS.

For XOT sessions established by HNAS the packet and window sizes are set as follows:

SVCs Non-GATE: The Call Request packet built by HNAS contains the FAC= string from the TYPE=MCH REMOTE FAC= operand or from TYPE=MXT REMOTE FAC= string associated with LU receiving the BIND that initiated the call. If the facilities string does not provide packet and window sizes the router will clear the call.

SVCs GATE: Packet and window sizes are provided in a prefix on the call request packet sent to HNAS by the CTCP. If the prefix packet size is zero, the value proved by this operand is used for the session. If the prefix window size is zero then 2 is used as the default window size.

PVCs: Packet and window sizes sent in the HNAS PVC SETUP packet are derived from the FAC= operand on the TYPE=MXT REMOTE addressed by the definition string in the PVC= operand. If no packet size is provided a default of 512 is used. If no window size is provided a default of 4 is used. If the sizes in the HNAS PVC SETUP packet do not match the sizes defined in the router then the router will reject the call.

[PORT={DYNAMIC|*number*|<u>3065</u>|<u>1998</u>}]

(DMY|XOT|XTP client)
(changed for V2R4M0)

Specifies the TCP port number for this remote router.

DYNAMIC specifies that the TCP port number will be set dynamically when the router connection is established. Additionally for XOT, indicates that HNAS takes the role of session listener while the XOT router takes the role of session initiator.

You may specify a ***number*** value between 0 and 65533 (65534 and 65535 are reserved). Currently, ***number*** values of 3065 (XTP) and 1998 (XOT) are the only values allowed. If a ***number*** value is not specified or if the specified value is invalid, a default value of **3065** will be used for an XTP client and **1998** will be used for an XOT client. These default port numbers are the ‘well known’ TCP port numbers for XTP and XOT peers, respectively.

Note: A single TYPE=XOT REMOTE definition statement can be used to specify multiple XOT clients. The VCLMT= operand value determines how many XOT clients will be generated. The PORT= and VCLMT= operand values together identify a pool of TCP sockets that will be used for the XOT clients.

Note: For TYPE=DMY REMOTE definition statements, the PORT= value you specify is used to identify the port on the router that will be the target of the PING *dmyname* console command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

```
[PROTOCOL={XTP|XOT}]
                    (DMY|DFX|DFS|DFL|SPU|MCH|XOT|XTP client)
                                        (restored for V2R4M0)
```

Specifies the TCPIP encapsulation protocol that is used for the REMOTE resource. The protocol is used primarily by TYPE=DMY REMOTE definition statements for the **PING *dmy-name*** console command.

XTP specifies that the TCPIP encapsulation protocol is XTP. This protocol is valid for TYPE=DMY|XTP REMOTEs.

XOT specifies that the TCPIP encapsulation protocol is XOT. This protocol is valid for TYPE=DMY|DFX|DFS|DFL|SPU|MCH|XOT REMOTEs.

```
[PVC={NONE|
      (vclmt,
      [{sluname[-{A|I}]|rmtname(4)||P||i}]
      [/{llcid|0}]
      [/{applid|255}]
      [/{lcn|0}]
      /ifname
      [/rmtname]
      [/mxtname],...)}]
                                        (MCH|XTP client)
                                        (new for V1R1M2)
                                    (changed for V2R3M0)
                                      (changed for V2R4M0)
```

Specifies either **NONE** or a Permanent Virtual Circuit count (***vclmt***) followed by a list that provides the following information for each PVC:

1) an SLU name (***sluname***) that is assigned to the PVC

2) an initial state for the SLU (**-{A|I}**) that is assigned to the PVC

3) a Logical Line Control (LLC) type (***llcid***) for the PVC

4) an application selection index (***applid***) for the PVC
For a TYPE=MCH REMOTE definition statements the following additional fields may be coded:

5) the Logical Channel Number (***lcn***) for the remote PVC
The ***lcn*** value is specified as a **hexadecimal number between 0 and FF** without the framing characters “X’’.

6) the serial interface name (***ifname***) that the real X25 link is attached to

7) a TYPE=XOT REMOTE definition statement name (***rmtname***) that will be used by HNAS to transmit an XOT PVC SETUP packet which will establish the PVC session

Note: HNAS does not initiate a PVC Setup request when the ***rmtname*** reference is omitted from the PVC= operand entry. When the ***rmtname*** is present, either end (the remote router or HNAS) can initiate the PVC Setup sequence.

8) the name of a TYPE=MXT REMOTE definition statement (***mxtname***) that provides packet and window size information which overrides the HNAS defaults. The PVC MXT is primarily used to override default vc packet and windows sizes that are provided in the HNAS initiated PVC Setup packet at vc initialization.

9) the position in the PVC= list provides the HNAS logical channel number (starting at 1) for the PVC.

The HNAS MCH name prefixed by SERIAL and the HNAS PVC logical channel number are used in an XOT router’s PVC definitions.

The first PVC operand list entry specifies that PVCs are not used (**NONE**) or specifies a PVC virtual circuit limit (***vclmt***).

If **NONE** is specified, PVC support will not be provided for this MCH.

If a ***vclmt*** value is specified, PVC support will be provided. The value coded indicates how many PVCs will be allowed for this MCH.

For **XOT PVCs (TYPE=MCH)**, HNAS PVC logical channel numbers (fourth (4th) suboperand (***lcn***) of each PVC entry) must match the PVC logical channel numbers define at the remote.

For **XTP PVCs (TYPE=XTP)**, HNAS PVC numbers map one-to-one with PVC Logical Channel Numbers (LCNs) on the router MCH (IFNUM=number). The PVC number is the 1-byte MCH LCN that is presented in the XTP PVC Information packet. Once established, PVC sessions, like their SVC counterparts, communicate using XTP packets that carry an interface number to identify the MCH on the router and a circuit number to identify the LCN on the MCH.

HNAS allocates and formats a Logical Unit Block (LUB) and Virtual Circuit Block (VCB) for each PVC identified in the PVC operand.

You may specify a ***vclmt*** value between 1 and 255. If a ***vclmt*** value is not specified or if the specified value is invalid, a default value of **NONE** will be used.

The ***sluname*** value you specify identifies a terminal session SLU that can be accessed by multiple host applications as an LUT1 device. The ***sluname*** value can be any valid assembler language symbol.

If an ***sluname*** value is omitted (/ or an LLCi is the first suboperand), a **default SLU name** is generated from the first four (4) characters of the REMOTE name (***rmtname***), padded on the right with a **#** or **£** characters (X'7B') as necessary, followed by the character 'P' and the 3-digit hexadecimal PVC operand mapping list index value (plus one).

Note: Each of the specified or generated ***sluname*** values must also appear on an APPL statement in the HNAS VTAM Application Major Node File (AMNF).

You may specify **-{A|I}** after the SLU name to provide the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When the SLU initial state is active HNAS will attempt to establish a VTAM session with the PLU. When the state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for communication with the PLU. HNAS will send a PVC SETUP packet to establish the PVC session with the router if the initial state is idle. A PVC SETUP received from the router will be accepted by HNAS (connected response returned) if the LU is offline.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

You may specify an ***llcid*** value of 0 for LLC0 (PCNE), 3 for LLC3 (QLLC) or 5 for LLC5 (PAD). If an ***llcid*** value is not specified (//) or if the specified value is invalid, a default value of **0** will be used.

The ***applid*** value is an index (relative to zero) into the APPLNAME= or LUNAME= operand lists. The APPLNAME= operand is used when the ***llcid*** value is 0 (PCNE), 3 (QLLC) or 5 (PAD).

You may specify an ***applid*** value between 0 and the APPLNAME= operand entry counts minus one (N'APPLNAME-1) or 255. An ***applid*** value of 255 indicates that the PVC is passive and that a host application will acquire (BIND) it. If an ***applid*** value is not specified (//) or if the specified value is invalid, a default value of **255** (passive) will be used.

Note: PVCs may not be used for CONSOLE or MCHSOL access. If an ***applid*** value is specified that selects an APPLNAME= operand entry for CONSOLE or MCHSOL, the PVC will be unusable.

Note: For QLLC PVCs, the real ***applid*** value comes from the SYSL= operand on the TYPE=SPU REMOTE definition statement. The ***applid*** value on this REMOTE definition statement is used as a place holder only but must be a valid value.

The ***mxtname*** value you specify identifies an MCH extension (MXT) that supplies overriding operands like FAC= for the PVC Setup packet. The name you specify for ***mxtname*** must appear in the name field of a TYPE=MXT REMOTE definition statement that appears elsewhere in the CDF. The ***mxtname*** value can be any valid assembler language symbol.

The PVC operand has been modified to allow an MXT to be associated with a PVC entry. The packet size and window size for the PVC will be extracted from the FAC= operand on the associated MXT via facility 42 and 43, respectively. For example:

```
PVC=(...,slunm/llc/applid/lcn/ifname/rmtname/mxtname,...)

mxtname REMOTE TYPE=MXT        ; typical network values
               FAC=(420707,    ; set packet size to 128
                    430202)    ; set window size to 2
```

In earlier releases, HNAS did not allow the PVC packet and window sizes to be specified in the CDF. The defaults set by HNAS are packet size=256 and window size=4 for both inbound and outbound flow. These values are carried in the PVC Setup packet sent by HNAS to start the PVC session.

If the values are rejected by the remote because the remote's configuration has different values then you should associate an MXT with the PVC entry and specify the correct packet and window sizes using the FAC= operand. The problem may occur intermittently because in almost all cases the router starts the PVC establishment process and HNAS will set the packet and window sizes to the values provided from the routers PVC Setup packet.

Note: Cisco ios serial interface parameters: '**x25 win|wout n', 'x25 ips|ops nnn**' and pvc 'pvc n xot ipaddr interface serial n/n mchname pvc n **packetsize nnn nnn windowsize n n**' are used to configure the Cisco router pvc packet and window sizes. Please refer to the XOT PVC Resource Definition sample in Chapter 3 for additional information.

Note: In this enhancement, the PVC packet size and window size are expressed using SVC Facility Parameter values normally used in Call Request packets. For PVC's the packet and window sizes are provided in the PVC Setup packet.

Note: For TYPE=XTP REMOTE definition statements, the ***lcn***, ***ifname*** and ***rmtname*** suboperands are invalid. If an ***mxtname*** suboperand is required, you must specify null values for the ***lcn***, ***ifname*** and ***rmtname*** suboperands as follows:

```
PVC=(...,slunm/llc/applid////mxtname,...)
```

Note: HNAS will dynamically adjust to the window and packet size values provided in the PVC Setup packet from the network router when the PVC setup is initiated from router which is typically the case. When HNAS initiates the PVC setup request and the X.25 network attached to the router does not support the provided window or packet size values, the router will reject the HNAS PVC Setup packet. In this case the HNAS provided window and packet size values will need to be specified to match the network values using an associated MXT.

For a TYPE=MCH REMOTE definition statement only, the following additional XOT PVC suboperands are allowed:

You may specify an ***lcn*** value as a hexadecimal number between 0 and FF without the framing characters "X''. The ***lcn*** value represents the Logical Channel Number that the remote

XOT router will use for the PVC. If an ***lcn*** value is not specified (//) or if the specified value is invalid, a default value of **0** will be used.

The ***ifname*** value you specify identifies a physical MCH on the remote router and is required. The interface name represents the MCH link that 'owns' the PVC LCN connection. The ***ifname*** values can be any text string up to 16 characters. in length.

If a slash (/) is required in the ***ifname*** text string, code a single hyphen (-) as a substitute character (slashes, by HNAS convention, are used as suboperand separator characters). If a hyphen is required, code two (2) hyphens consecutively (--).

For example, if SERIAL0/1 is required as an ***ifname*** value, specify SERIAL0-1. If SERIAL0-1 is required as an ***ifname*** value, specify SERIAL0--1. If an ***ifname*** value is not specified or if the specified value is invalid, HNAS will terminate. The ***ifname*** value is required by HNAS. It is carried in the XOT PVC Setup packets that are exchanged between HNAS and the router.

The following are eamples of valid Cisco interface names:

SERIAL1-0-1 which is saved as SERIAL1/0/1
SERIAL1-3-1 which is saved as SERIAL1/3/1
SERIAL1-0:0-1 which is saved as SERIAL1/0:0/1
SERIAL1-3:0-1 which is saved as SERIAL1/3:0/1

The ***rmtname*** value you specify identifies a pool of TCP/IP sockets that are used for outbound XOT client connections. The name you specify for ***rmtname*** must appear in the name field of a TYPE=XOT REMOTE definition statement that appears elsewhere in the CDF. The ***rmtname*** values can be any valid assembler language symbol. If a ***rmtname*** value is not specified or if the specified value does not identify a TYPE=XOT REMOTE, it is ignored. In this case, the PVC connection is assumed to be initiated by the router.

The pool of TCP/IP sockets is created from the IPADDR=***a.b.c.d***, PORT=**1998** and VCLMT=***count*** operand values on the named TYPE=XOT REMOTE definition statement. This pool is referred to as a ***shared socket pool*** because it can be used for both outbound and inbound connections. The IPADDR= operand identifies a specific router in the IP network. The PORT= operand must specify TCP port number 1998 which is the 'well known' XOT port number. The VCLMT= operand value defines that size of the pool. For example, IPADDR=192.40.60.4,PORT=1998,VCLMT=4 specifies that four (4) sockets are reserved in the ***shared socket pool*** for the given XOT router.

If the number of PVC operand list entries is less than the ***vclmt*** value that is in effect, default values will be assumed for the unspecified entries except for the ***ifname*** value which is required when TYPE=MCH is specified.

You may specify from 1 to ***vclmt*** PVC operand list entries.

Example:
```
TYPE=XTP                     ; XTP CLIENTS
GATE=GENERAL                 ; GATE FUNCTION REQUIRED
CONNECT=NO                   ; NO GATEFC
```

```
PAD=TRANSP                      ; XPAD SUPPORT REQUIRED
PVC=(8,                         ; ALLOW 8 PVS
0/0,4/0,4/1,4/2,0/1,            ; SUPPLY MAPPING FOR PVC 1-8
5/2,4/255,4/255)                ; DEFAULT SLU NAMES ASSUMED
LUNAME=(CTCPSLU0,               ; SUPPLY CTCP SLU NAMES
        CTCPSLU1,
        CTCPSLU2)
APPLNAME=(TSO,CICS,USER)        ; IDENTIFY NON-CTCP APPS
```

LLC and CTCP or non-CTCP application selection for PVCs is accomplished as follows:

1.) The PVC operand list is first examined when an XTP PVC Information packet is received. This generally occurs immediately after the router first connects to HNAS.

2.) If the PVC entry contains an ***applid*** value of 255, HNAS establishes the LLC for the PVC then waits for a host application to acquire (BIND) it.

3.) If the PVC entry contains an ***applid*** value less than 255, HNAS examines the ***llcid*** value to see if the LUNAME= or APPLNAME= operand will be used for application selection. The LUNAME= operand is used for LLC4 and the APPLNAME= operand is used for LLC0 or LLC5. For the LUNAME= operand, if the selected CTCP SLU is not bound or for the APPLNAME= operand, if the selected host application is not active (REQSESS rejected), PVC system select processing is repeated after a forced delay.

Note: Since the TRAN= operand is ignored for LLC0 sessions, data that passes between an EBCDIC application like TSO and the remote device must be EBCDIC end-to-end. HNAS will transfer data transparently on input and output.

Note: For LLC4 sessions, the PVC is connected to the CTCP data session SLU. There is no activity on the CTCP control session SLU.

4.) In the example above, PVC1 is assigned LLC0 and selects TSO, PVC2 is assigned LLC4 and selects CTCPSLU0, PVC3 is assigned LLC4 and selects CTCPSLU1, PVC4 is assigned LLC4 and selects CTCPSLU2, PVC5 is assigned LLC0 and selects CICS, PVC6 is assigned LLC5 and selects USER and finally, PVC7 and PVC8 are assigned LLC4 and wait for acquisition.

HNAS XOT router differences in PVC operation

1.) The router will reject (with STATUS=18) an HNAS PVC SETUP packet which has packet and window sizes that do not match those in its own definition tables. HNAS will accept a PVC SETUP packet with any packet and window sizes. As a general rule the router and HNAS PVC definitions should always match unless HNAS is configured to never send PVC Setups which will cause HNAS to set it's window and packet size based upon the values provided in the PVC Setup packet from the remote router.

2.) The router will send PVC SETUPs even if the serial interface is inoperative. HNAS will not send PVC SETUPs if the TYPE=XOT REMOTE is offline.

3.) The router will reject (with STATUS=1B) a PVC SETUP from an IP address not in the router configuration. HNAS will accept a PVC SETUP from any IP address if a TYPE=MCH REMOTE with a name matching the SETUP's responder field is located and if an LCN is defined in HNAS which matches the LCN in the PVC setup sent by the router.

4.) The router will reject (with STATUS=17) a PVC SETUP packet received when the PVC session is already operating. HNAS terminates the operating session and accepts the new SETUP.

HNAS has two primary timers that control the operation of PVCs

An outbound PVC SETUP is scheduled by MCH one minute timer logic when a PVC generated with a TYPE=XOT remote pointer and no TCP/IP session is detected.

HNAS PVC connect timers are used to cause HNAS to start the LU connect sequence (OPEN ACB, SETLOGON, REQSESS) which establishes the HNAS SLU/PLU VTAM session. The timer (VCPVCTOD) is not started unless the PVC SETUP sequence has completed (i.e. PVC to PVC session active and in data transfer mode P4D1).

The following connect timer values are used:

60 seconds:	MCH timer logic is on a one minute boundary and a PVC with a closed ACB is detected.
240 seconds:	MCH timer logic is on a one minute boundary and a PVC with an open ACB, an idle LU and a known PLU name is detected. This status indicates that the HNAS REQSESS operation failed.
240 seconds:	LLC0/LLC5 PLU sent UNBIND to HNAS.
120 seconds:	REQSESS failed (session rejected by PLU).
240 seconds:	ACB open fails (e.g. no active Application Major Node for HNAS SLU name).

An XOT PVC definition example is available in Chapter 3 section XOT PVC Resource Definitions under the heading Example HNAS CDF XOT PVC Definition.

[PWPROT={<u>NO</u>|YES|YESWOCC}]

(MCH|XTP client)

Specifies whether the host will be allowed to control password protection by toggling the ECHO ON|OFF X.29 parameter. The host sets ECHO OFF by placing an **Inhibit Presentation (INP) character** at the end of an FMD PIU (INP=X'24' when **not** TRAN=NO or INP=X'12' when TRAN=NO). The host sets ECHO ON by placing an **Enable Presentation (ENP) character** at the start of an FMD PIU (ENP=X'14' when TRAN=YES|NO). PAD=INTEG must also be specified.

NO specifies that HNAS will treat INP and ENP characters as normal data characters thus preventing the host from manipulating the ECHO ON|OFF X.29 parameter.

YES specifies that HNAS will react to the INP character when it is in the last byte position of an FMD PIU and the ENP character when it is the first byte position of an FMD PIU.

HNAS will send the INP or ENP character in an XTP or XOT non-Qualified Data packet followed by a 'set parameters' XTP or XOT Qualified Data packet with X.29 parameter 2 set to 0 (ECHO OFF) for INP or 1 (ECHO ON) for ENP.

YESWOCC specifies that HNAS will react to the INP character when it is in the last byte position of an FMD PIU and the ENP character when it is the first byte position of an FMD PIU.

HNAS treats the YESWOCC and YES values the same except that YESWOCC also causes HNAS to delete the INP and ENP characters from the output data stream.

If the PWPROT operand is not specified or if the specified value is invalid, a default value of **NO** will be used.

[SUBADDR={NO|YES}]

(MCH|XTP client)

Specifies whether the subaddress digit (defined as the last digit in the *called* DTE address field) from an **inbound** Call Request packet will be used for Logical Line Control (LLC) selection.

NO specifies that the subaddress digit will not be used for LLC selection.

YES specifies that the subaddress will be used for LLC selection. The LLC0=, LLC3=, LLC4= and LLC5= operands identify values that can set the LLC type. GATE=GENERAL must also be specified. If the subaddress digit is not in the LLCi operands then CUD0 is used to determine the LLC. If an LLC cannot be determined the call is cleared.

If the SUBADDR= operand is not specified or if the specified value is invalid, a default value of **NO** will be used.

[SUBD=(*value*1,...,*value*n)]

(MCH|XTP client)
(changed for V2R2M0)

Specifies a lookup list that is used to map the SUBD digits from XTP or XOT **inbound** Call Request packets to CTCP= operand entries.

The SUBD= operand specifies a list of values that are used for mapping subaddress digits to CTCP= operand list entries. When an XTP or XOT inbound Call Request packet is received, the subaddress digits are tested against the values in the SUBD= operand list. If a match occurs, the position of the matched SUBD value in the SUBD= operand list is used as an index into the CTCP= operand list. GATE=GENERAL and CONNECT=NO|SUBD must also be specified.

Note: For HNAS releases prior to V2R2M0, the SUBD= operand could only be used for GATEFC which was invoked by specifying CONNECT=SUBD. This restriction has been removed so that the SUBD= operand, like the CUD0= operand, can also be used for normal GATE which is invoked by specifying CONNECT=NO.

Specifying CONNECT=NO and SUBD=(aa,...,nn) allows subaddress digits from a Call Request packet to be used to override the CTCP selected for an inbound non-Fast Connect GATE (standard GATE) call. Non Fast Connect GATE CTCP selection operates as follows:

If LLC4 (GATE) is selected by a subaddress digit, the first CTCP in LUNAME= is selected.

If LLC4 is selected by a CUD0 value, the CTCP is selected by CUD0 in conjunction with the CTCP= list or a default table of CUD0 values (see CUD0= in this chapter).

After LLC4 has been set and a CTCP has been selected by one of the above two methods, HNAS checks to see if SUBD=(aa,...,nn) was coded. If SUBD= was coded, the subaddress digits are checked against the entries in the SUBD= operand list. If a match is found, the corresponding entry (by position) in the CTCP= operand list provides a new index in the LUNAME= list of CTCPs. The CTCP selected replaces the CTCP selected by previous processing. If no match is found or if the selected CTCP= operand entry is not a CTCP index, the originally selected CTCP is used.

Specifying CONNECT=SUBD and SUBD=(aa,...,nn) indicates that the MCH is a Fast Connect GATE MCH (all calls are GATEFC) and that CTCP selection is to be made using subaddress digits (only).

Note: When a one (1) digit SUBD value is specified (e.g., 9), only the last digit of the ***called*** DTE address field is tested for a subaddress match. When a two (2) SUBD value is specified (e.g., 09), the last two (2) digits of the ***called*** DTE address is tested for a subaddress match. Since the entire ***called*** DTE address is given in packed decimal format, SUBD values can range from 0 to 99.

You may specify a ***value*i** value between 0 and 99. If a ***value*i** value is not specified or if the specified value is invalid, it is ignored.

You may specify from 1 to 100 SUBD= operand list entries. The number of entries in the SUBD= and CTCP= operands must be the same.

An example for CONNECT=SUBD processing can be found on page 4-76 of this document.

Primary Format

```
[SVC0={NONE|(vclmt,
            [{sluname[+gluname][-{A|I}]|rmtname(4)||0||i}]
            [/{X||idnum1|dteaddr1/&lt;rmtname1&gt;}
               [-{X||idnum2|dteaddr2/&lt;rmtname2&gt;}
                  [-{X||idnum3|dteaddr3/&lt;rmtname3&gt;}]]
                     ||{T|O|I}[{applid|0}]}]
            [/mxtname]
            [/cud],...)}]
```

or, when you want to control SLU name generation and no additional parameters are required

Alternate Format

```
[SVC0={NONE|(pfxlu,sfxst,vclmt)}]
```

(MCH|XTP client)
(new for V1R1M3)
(changed for V2R1M1)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

For the Primary Format

Specifies either **NONE** or an LLC0 Switched Virtual Circuit count (***vclmt***) followed by a mapping list that identifies, for each LLC0 SVC, (1) an SLU name (***sluname***) that is assigned to the SVC and optionally a generic SLU name (***gluname***) for use by VTAM acquire operations (PLU callout), (2) a remote DTE identifier specified as a DTE address (***dteaddr***), a hex IDNUM value (**X*idnum***) or the name of a TYPE=MXT REMOTE definition statement (***<rmtname>***) whose DTEADDR= operand identifies the remote DTE, (3) a connection identifier ({**I**|**O**|**T**}) that determines the call setup procedure, (4) an application selection index (***applid***), (5) the name of a TYPE=MXT REMOTE definition statement (***mxtname***) that provides overriding operands and (6) call user data (***cud***) that will override any CUD= operand data and be placed in the outbound Call Request packet that HNAS creates when the SLU is bound (SLU is marked for callout (**O**|**T**)). GATE=NO|GENERAL and CONNECT=NO must also be specified.

The first SVC0=operand list entry specifies that LLC0 (PCNE) SVCs are not used (**NONE**) or specifies an LLC0 SVC virtual circuit limit (***vclmt***).

If **NONE** is specified, LLC0 SVC support will not be provided for this MCH.

If a ***vclmt*** value is specified, LLC0 SVC support will be provided. In this case, the second SVC0= operand list entry is for the first LLC0 SLU, the third SVC0= operand list entry is for the second LLC0 SLU, and so on. HNAS SVC0= operand indices are independent of the SVC Logical Unit Numbers (LCNs) that are used by the router.

Once established, XTP SVC sessions communicate using XTP packets that carry an interface number to identify the MCH on the router and a circuit number to identify the LCN on the MCH. XOT SVC sessions communicate using XOT packet that carry an LCN. The MCH link is identified by the TCP/IP socket itself.

HNAS allocates and formats a Logical Unit Block (LUB) and Virtual Circuit Block (VCB) for each SVC identified in the SVC0= operand.

The ***vclmt*** value you specify indicates how many LLC0 SVCs will be allowed for this MCH.

You may specify a ***vclmt*** value between 1 and 511. If a ***vclmt*** value is not specified or if the specified value is invalid, **an error message is generated if** GATE=NO, PAD=NO and SVC3=NONE are specified **or if** GATE=GENERAL, SUBADDR=YES and LLC0=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,80,...) are specified. Otherwise, **NONE** will be used.

Note: We suggest that you specify **NONE** for the SVC*i*= operands that you do not plan to use and a ***vclmt*** value for those that you do plan to use.

The ***sluname*** value you specify identifies an PCNE terminal session SLU that can be accessed by multiple host applications as an LUT1 device. The ***sluname*** value can be any valid assembler language symbol.

The ***gluname*** value (if specified) identifies a generic SLU name that will be passed to VTAM (SETLOGON OPTCD=GNAMEADD) when the ACB for ***sluname*** is opened. This allows PLUs to ACQIUIRE the HNAS resource (for callout operations) using ***gluname.*** Generic resources must also be defined in VTAM.

Note: ***sluname+gluname*** support was introduced into 240 as Enhancement APAR 2400084.

If an ***sluname*** value is omitted (/), a **default SLU name** is generated from the first four (4) characters of the REMOTE name (***rmtname***), padded on the right with **#** or **£** characters (X'7B') as necessary, followed by the character '0' and the 3-digit hexadecimal SVC0= operand index value (plus one).

Note: Each of the specified or generated ***sluname*** values must also appear on an APPL statement in the HNAS VTAM Application Major Node File (AMNF)**.**

You may specify **-{A|I}** after the SLU name to provide the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When the SLU initial state is active, the SLU is conditioned to accept or solicit LLC0 (PCNE) SVC calls. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for LLC0 SVC connections.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

For all SVC0= operand entries, 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 be enclosed in non-equivalence symbols (<>) and must name a TYPE=MXT|SVC REMOTE definition statement. 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.

Inbound **_dteid_i** list processing:

When the **_dteid_i** list entries are identified as **I**nbound (the connection identifier is omitted or is specified as **I**) or **T**woway (the connection identifier is specified as **T**), the following rules are used to allocate an SLU using the **_dteid_i** values:

When a **_dteaddr_i** value is specified, it must match the **_calling_** DTE address in the incoming Call Request packet (for the number of digits specified by **_dteaddr_i**) in order for the associated SLU to be allocated. The **_dteaddr_i** represents the physical DTE at the X.25 network entry point which is the only DTE that will be allowed to use the SLU. If the **_dteaddr_i** value does not yield a match, the next **_dteid_i** list entry is examined.

When an **X_idnum_i** value is specified, it must match the Call User Data (starting at byte 1) in the incoming Call Request packet (for the number of bytes specified by **X_idnum_i**) in order for the associated SLU to be allocated. If the **X_idnum_i** value does not yield a match, the next **_dteid_i** list entry is examined. If the CUD0SELECTSLU option is specified, the compare for an **X_idnum_** value starts at CUD0 instead of CUD1.

Note: For NPSI emulation of CUD IDNUM data which is 5-digits in length, you would actually code 6-digits for **X_idnum_i**, the last being zero (0). For example, if the CUD for an incoming PCNE Call Request packet is C0100020, the IDNUM portion is 10002. In the SVC0 entry, you would specify the **X_idnum_i** value as X100020 since the compare is done on CUD bytes 1-3.

Note: The same **X_idnum_i** value may be specified for different SLUs on different MCHs. The MCH is selected via RTEIN= mapping. Once the MCH is selected, the **X_idnum_i** value is then used to select the SLU.

When a **<_rmtname_i>** value is specified, it must identify a TYPE=MXT|SVC REMOTE definition statement whose DTEADDR=**_dteaddr_i** value will be used to provide a DTE address for **_calling_** DTE address matching. If no **_dteaddr_i** value is given on the MXT|SVC or if the specified **_dteaddr_i** value does not yield a match, the next **_dteid_i** entry is examined.

If all **_dteid_i** list values have been tested without yielding a match, the SLU is not allocated and the call is cleared.

If no **_dteid_i** list is given, the SLU is allocated on a first come, first served basis.

If you specify a connection identifier character of **I|T** following the **_dteid_i** list, you may also specify a APPLNAME= operand index value (**_applid_**) following the **I|T**. You may specify an **_applid_** value between 0 and N’APPLNAME-1 (the APPLNAME= operand entry count minus one). If an **_applid_** value is specified, HNAS will use it to select a PLU name for a REQSESS request. This feature allows you to dedicate an SLU to a specific application (PLU) whose name appears in the APPLNAME= operand list. If an **_applid_** value is not specified, the SYSL= operand will be used to select an application. If the SYSL= operand is also not specified, a default **_applid_** value of 0 will be used to select the first APPLNAME= operand entry.

Outbound **_dteid_i** list processing:

When the **_dteid_i** list entries are identified as **O**utbound (the connection identifier is specified as **O**) or **T**woway (the connection identifier is specified as **T**), the following rules are used to establish a connection to the remote DTE using the **_dteid_i** values in an outbound Call Request packet:

When a **_dteaddr_i** value is specified, it is used as the **_called_** DTE address for the outbound call. The **_calling_** DTE address, facilities and call user data come from the DCEADDR=, FAC and CUD= operands on either the associated MXT (see **_mxtname_** without <> below) or the root MCH.

When a **<_rmtname_i>** value is specified, the DTEADDR=**_dteaddr_i** and DCEADDR=**_dceadd_i** operands on the named MXT|SVC are used to provide the **_called_** and **_calling_** DTE addresses, respectively, for the outbound call. The FAC= and CUD= operands on the same MXT|SVC provide the facilities and call user data for the outbound call.

Note: When an MXT|SVC is specified as a *rmtname*i in place of a *dteaddr*i value, it becomes the REMOTE that is associated with the SLU for the duration of the LU/VC session. In this case, it will be the REMOTE that provides the *called* DTE address (DTEADDR= operand), *calling* DTE address (DCEADDR= operand), facilities data (FAC= operand) and call user data (CUD= operand). These four operands provide all the information necessary to create a unique Call Request packet. After APAR 2400069, OPTIONS=T21=secs can be specified on the MXT to specify the number of seconds HNAS will wait for a response (Call accept or Clear) to an outbound Call Request.

For PAD calls that require PADPARM= values to override those on the root MCH, an MXT rather than an SVC should be used in place of a *dteaddr*i value. This is because the SVC REMOTE does not support the PADPARM= operand.

If PADPARM= values are not required, we recommend using an SVC rather than an MXT for call processing. The SVC control block uses less memory than the MXT. The SVC control block is X'178' (376) bytes in length while the MXT control block is X’480’ (1152) bytes in length.

Note: <*rmtnamei*> support was introduced into 240 as Enhancement APAR 2400003.

The purpose in providing multiple callout DTE addresses is to allow HNAS to retry failed outbound call requests. Make sure that all ***dteaddr*i** values pass the DTE address filter that you have specified for the RTEOUT operand on the HOME TYPE=XOT LOCAL definition statement. This will permit HNAS to allocate a TCP socket for the XOT connection. Each ***dteaddr*i** will be used, in turn, as the ***called*** DTE address in the outbound Call Request packet that HNAS creates when the SLU is bound if the call is cleared by the remote DTE.

For example, if 3 DTE addresses are specified for an SLU, ***dteaddr1*** will be used when the SLU is initially bound by the PLU. If HNAS receives a Clear Request packet as the response to its Call Request packet Call Request times out (nothing received), a new call will be attempted using ***dteaddr2***. If this call also fails, a new call will be attempted using ***dteaddr3***. If this call also fails, the PLU will be notified via an UNBIND request. If a Call Accepted packet is received as a response to any of the Call Request packets, normal processing will continue.

Example: `SVC0=(...,MCH10020/X900060-20360009-<MTX0>T2/MXT1/C000000,...`

The ***mxtname*** value you specify after the DTE address list identifies an MCH extension (MXT) that supplies overriding operands like CUD=, DCEADDR= and/or FAC= for callout and/or LOGTAB= and USSTAB= for callin. The name you specify for ***mxtname*** must appear in the name field of a TYPE=MXT REMOTE definition statement that appears elsewhere in the CDF. The ***mxtname*** value can be any valid assembler language symbol.

The CUD=, DCEADDR=, FAC=, LOGTAB=, PARPARM= and/or USSTAB= operands that are specified on the named TYPE=MXT REMOTE definition statement can supply different values than those coded on this REMOTE definition statement. If any or all of the callout operands are omitted from the named TYPE=MXT REMOTE definition, data is taken from the corresponding operand on this REMOTE definition.

Although the same TYPE=MXT REMOTE definition statement can be used for callout or callin SVCs, we recommend that you specify different TYPE=MXT REMOTEs for each type.

For SVC0= operand entries identified as **O**utbound (the connection identifier is specified as **O**) or **T**woway (the connection identifier is specified as **T**), you may specify a ***cud*** value that will be placed in the Call User Data field of the outgoing Call Request packet that HNAS creates. This ***cud*** value overrides the CUD= operand value from this REMOTE definition statement and from the TYPE=MXT REMOTE definition statement that is identified by ***mxtname***. If the CUD= operand is also omitted, a default value of C0000000 will be used.

If specified, the SVC0= ***cud*** value must be **exactly 4 bytes** (8 digit pairs) in length. If you require more or less than 4 bytes of call user data, you will have to code the CUD= operand on the TYPE=MXT REMOTE definition statement identified by ***mxtname*** or on this TYPE=MCH REMOTE definition statement.

Note: If the SLU entry is being used to connect to another HNAS or a NPSI FEP, the 4-byte ***cud*** value can be thought of as a protocol ID byte (**CUD0**) followed by a 5-digit (3-byte with zero pad digit on right) **X*idnum*** value. If you are migrating to HNAS from NPSI and need to use the IDNUM= value from the NPSI Switched Major Node PU/PATH statements, the value you should code for **X*idnum*** is the NPSI IDNUM= value plus one (1). The **CUD0** value should be **C0** for PCNE. NPSI takes the value that is specified for IDNUM= at the end of the

DIALNO= operand on the PATH statement and adds one (1) to it before sending it to the receiving side. HNAS takes the value you code for ***cud*** (**CUD0** plus **X*idnum***) and sends it as is. For example:

For NPSI:

```
PCLFTPU  PU    ADDR=C1,IDBLK=003,IDNUM=01225,PUTYPE=1,MAXPATH=1,
               SPAN=(SPANSP),
               ISTATUS=ACTIVE,MAXDATA=2540,VPACING=(1,1),
               DISCNT=YES
         PATH  DIALNO=3151858800000*01225,
               GRPNM=D25SGSSA,GID=1,PID=1
PCLFTLU  LU    LOCADDR=1
```

For HNAS:

```
         REMOTE ...
                 :
                SVC0=(vclmt,
                       :
                       PCLFTLU/dteaddr1-...O/mxtname/C0012260,
                       :
```

Call Request Parameter Hierarchy Matrix

PARAMETER	When *dteid*i=<*rmtname*i>	When *dteid*i=*dteaddr*i
***called* DTE address**	**DTEADDR= from <*rmtname*i>**	***dteaddr*i**
***calling* DTE address**	**DCEADDR= from <*rmtname*i>**	**1) DCEADDR= from *mxtname* if defined** **2) DCEADDR= from root MCH**
facilities data	**FAC= from <*rmtname*i>**	**1) FAC= from *mxtname* if defined** **2) FAC= from root MCH**
call user data	**CUD= from <*rmtname*i>**	**1) *cud* from SVC0 operand if defined** **2) CUD= from *mxtname* if defined** **3) CUD= from root MCH**

Note: If the operand at the bottom of the hierarchy list for a particular Call Request parameter is also undefined (null), the corresponding Call Request field will contain zeros.

If a specified SVC0= suboperand is in error, a default value will be substituted.

If the number of SVC0= operand list entries is less than the ***vclmt*** value that is in effect, default values will be assumed for the unspecified entries.

You may specify from 1 to ***vclmt*** SVC0= operand list entries.

```
Example: GATE=GENERAL                     ; GATE FUNCTION REQUIRED
         CONNECT=NO                       ; NO GATEFC
         SUBADDR=YES                      ; SUBD SETS LLC
         LLC0=(0,1,9)                     ; SUPPLY SUBD VALUES
         SVC0=(2,                         ; ALLOW 2 LLC0 SVCS
               SVC0SLU1/3026001I,         ; SUPPLY LLC0 SVC SLU NAMES/ADDR
               SVC0SLU2/3026001O)
         SYSL=(CUD4=41/0,CUD4=42/1)       ; SUPPLY APPL MAPPING
         APPLNAME=(TSO,CICS)              ; IDENTIFY NON-CTCP APPS
```

For the Alternate Format

Specifies either **NONE** or an SLU name prefix value (***pfxlu***) followed by a suffix start value (***sfxst***) followed by an LLC0 Switched Virtual Circuit count (***vclmt***).

The primary and alternate specifications are both be accepted. The alternate specification is appropriate when no additional suboperands need to be associated with each SLU name. For the alternate specification:

The ***pfxlu*** value must be the first SVC0= suboperand and may be any valid assembler language symbol up to 7-characters in length starting with either an alpha character (A,B,C,...,Z) or an accepted special character (@, #, $ or %). This suboperand is REQUIRED to indicate that the alternate format is being used.

The ***sfxst*** value must be the second SVC0= suboperand and must be a hexadecimal number (without the framing characters X'') between 0 and F when the ***pfxlu*** length is 7, between 0 and FF when ***pfxlu*** length is 6, ..., between 0 and FFFFFFF when the ***pfxlu*** length is 1. If ***sfxst*** is omitted (,,), a default value of 0 will be used.

The ***vclmt*** value must be the third SVC0= suboperand and must be a decimal number between 0 and 511 (the SVC0= array size). This suboperand is REQUIRED. The SLU names that HNAS generates from the ***pfxlu***, ***sfxst*** and ***vclmt*** values will always be 8-characters in length with zero (0) pad characters added as required between the last ***pfxlu*** character and the suffix value.

Examples:

If SVC0=(XX,1,3) is specified, the generated SLU names would be XX000001, XX000002, XX000003.

If SVC0=(XXXXX,0,3) is specified, the generated SLU names would be XXXXX000, XXXXX001, XXXXX002.

SVC0= Operand Run Time Processing

Application (PLU) selection for LLC0 SVCs is accomplished as follows:

1. For callout entries in the SVC0= operand list, HNAS will set LLC0 and generate an **outbound** Call Request packet when the SVC0 SLU is acquired (bound). No additional system selection is required because the SVC0 SLU is 'owned' by the binding host application.

2. For callin entries in the SVC0= operand list, HNAS will wait for the arrival of an **inbound** Call Request packet. When an LLC0 call is received (as determined by CUD0 or a subaddress digit) the SVC0= operand is used for SLU selection. SLU allocation is a two pass operation performed as follows:

   Pass 1 (scan entries with idnum or DTE address values)
   If an SVC0= operand entry contains a hex **X*idnum*** value, it is compared against the IDNUM value from the **CUD** field of the inbound Call Request packet (byte 1-n, where n is the number of bytes you code for ***idnum***). If a match occurs, the SLU is allocated to the VC. If no match occurs, the next SVC0= operand entry is examined. If the CUD0SELECTSLU option is coded the compare starts at CUD0.

   If an SVC0= operand entry contains a decimal callin ***dteaddr*** value (designated with **I** as its connection identifier), it is compared against the ***calling*** DTE address from the inbound Call Request packet (**for the number of digits specified by *dteaddr***). If a match occurs, the SLU is allocated to the VC. If no match occurs, the next SVC0= operand entry is examined. This form of SLU selection guarantees that the remote DTE will always access host applications using the same SLU resource.

   Pass 2 (scan entries without idnum or DTE address values)
   If no match is found in the SVC0= entries with idnum or DTE address values then the remaining SVC0= entries are searched for an available SLU. If one is found it is allocated to the VC.

   If all SVC0= operand entries have been examined and no SLU can be allocated, the call is cleared.

3. After SLU selection has been completed, the SYSL= operand is used for application (PLU) selection. The SUBD= and/or CUD*i*= suboperands of the SYSL= operand specify mapping values and indices into the APPLNAME= operand list.

   If the SYSL= operand is omitted or if no match occurs in the SYSL= operand list, the first APPLNAME= operand entry will be used as the default application. If the selected host application is not active (REQSESS rejected) or if the Console Subsystem is selected and it is unavailable, the call is cleared.

   If an APPLNAME= operand index (***applid***) is coded for the SLU after the **X*idnum*** or ***dteaddr*** delimiter character **I**, this index used to locate an application in the APPLNANE= operand list. Since no user input is required for application selection, the SLU is effectively dedicated to one application (PLU).

4. In the example above, a CUD4 value 41 results in the selection of TSO and a CUD4 value of 42 results in the selection of CICS.

```
[SVC3={NONE|ALLOW|
          (vclmt,
           [spuname]
           [/dteaddr||{I|O}]
           [/mxtname],...)}]
```

(MCH client only)
(new for V2R2M0)
(changed for V2R3M0)

Specifies either **NONE**, **ALLOW** (any number) or an LLC3 Switched Virtual Circuit count (***vclmt***) followed by a mapping list that identifies, for each LLC3 SVC, (1) the name of a TYPE=SPU REMOTE definition statement (***spuname***) that is assigned to the SVC, (2) a DTE address (***dteaddr***) that identifies the remote DTE, (3) a connection identifier ({**I|O**}) that determines the call setup procedure and (4) the name of a TYPE=MXT REMOTE definition statement (***mxtname***) that provides overriding operands.

The collection of TYPE=SPU REMOTE definition statements in the CDF forms a pool of SPUs. An SPU is allocated from this pool when an LLC3 (QLLC) SVC activates either specifically from an SVC3= operand list entry or dynamically by selecting the first available SPU in the pool.

The ***vclmt*** value is required to accept LLC3 connections, however, if no ***spuname*** values are specified, SPU allocation is done dynamically from the SPU pool based on IDBLK/IDNUM matching.

A ***vclmt*** value is required even if no SPU names are specified. The MCH will not allow QLLC connections unless an LLC3 ***vclmt*** value is given. This can be confusing when SPU allocation is done using IDBLK/IDNUM match only. Any MCH defined for QLLC support can access any SPU in the CDF. Rather than requiring that SVC3=1 be specified to mark the MCH for QLLC support, **SVC3=ALLOW** will now be accepted. This new keyword indicates that the MCH allows any number of QLLC connections.

For SVC3= operand entries identified as **I**nbound (the connection identifier is omitted or is specified as **I**) and an ***spuname*** value is specified with an associated ***dteaddr*** value, the specified SPU will be allocated to a VC if the ***calling*** DTE address in the incoming Call Request packet matches the ***dteaddr*** value. If no match occurs for any ***dteaddr*** value in the SVC3= operand list, dynamic SPU allocation will be performed.

For SVC3= operand entries identified as **O**utbound (the connection identifier is specified as **O**) and an ***spuname*** value is specified with an associated ***dteaddr*** value, the specified SPU will be allocated to a VC and a Call Request packet will be created using the ***dteaddr*** value as the ***called*** DTE address. HNAS will initiate an SVC connection to the associated SPU automatically when the **OPTIONS=MCHTMR=*value*** expires for the MCH. Initiation of the outbound QLLC call will continue until a Call Accept packet is received. An accepted call remains active until cleared by the remote SPU. Following a clear, HNAS will attempt to reestablish the connection to the SPU after a forced delay.

Note: If an SPU is defined with a CLOTINITYP= value and it is also referenced in the SVC3= operand for a TYPE=MCH REMOTE definition statement as a callout SPU (the character O follows the associated DTE address), CLOTINITYP= call initiation will supersede SVC3= call-out initiation. See the description of the OPTIONS=CLOTINITYP= operand for additional information.

Note: The same ***spuname*** can be specified in the SVC3= operand lists on more than one (1) TYPE=MCH REMOTE definition statement. SPUs are always allocated on a first come, first served basis.

The first SVC3=operand list entry specifies that LLC3 (QLLC) SVCs are not used (**NONE**) or specifies an LLC3 SVC virtual circuit limit (***vclmt***).

If **NONE** is specified, LLC3 SVC support will not be provided for this MCH.

If a ***vclmt*** value is specified, LLC3 SVC support will be provided. In this case, the second SVC3= operand list entry is for the first LLC3 SPU, the third SVC3= operand list entry is for the second LLC3 SPU, and so on. HNAS SVC3= operand indices are independent of the SVC Logical Unit Numbers (LCNs) that are used by the router.

Once established, XOT SVC sessions communicate using XOT packets that carry an LCN. The MCH link is identified by the TCP/IP socket itself.

HNAS allocates and formats a Virtual Circuit Block (VCB) for each SVC identified in the SVC3= operand.

The ***vclmt*** value you specify indicates how many LLC3 SVCs will be allowed for this MCH.

You may specify a ***vclmt*** value between 1 and 255. If a ***vclmt*** value is not specified or if the specified value is invalid, **an error message is generated if** GATE=NO, PAD=NO and SVC0=NONE are specified **or if** GATE=GENERAL, SUBADDR=YES and LLC3=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,83,...) are specified. Otherwise, **NONE** will be used.

Note: We suggest that you specify **NONE** for the SVC*i*= operands that you do not plan to use and a ***vclmt*** value for those that you do plan to use.

The ***spuname*** value you specify identifies a TYPE=SPU REMOTE definition statement that appears elsewhere in the CDF. The ***spuname*** value can be any valid assembler language symbol.

If an ***spuname*** value is omitted (/), the SVC3= operand entry will be ignored.

You may specify a ***dteaddr*** value up to fifteen (15) decimal digits in length followed by a connection identifier (**I** for callin). If a connection identifier is not specified, **I** is assumed. The ***dteaddr*** represents the physical DTE at the X.25 network entry point which is the only DTE that will be allowed to use the SPU.

For callin, the ***dteaddr*** you specify must match the ***calling*** DTE address in the Call Request packet in order for the associated SPU to be allocated. If no ***dteaddr*** is given, the SPU is allocated on a first come, first served basis.

For callout, the ***dteaddr*** you specify will become the ***called*** DTE address in Call Request packet that HNAS creates.

The ***mxtname*** value you specify identifies an MCH extension (MXT) that supplies overriding operands like CUD=, DCEADDR= and/or FAC= for callout. The name you specify for ***mxtname*** must appear in the name field of a TYPE=MXT REMOTE definition statement that appears elsewhere in the CDF. The ***mxtname*** value can be any valid assembler language symbol.

The CUD=, DCEADDR=, FAC=, LOGTAB= and/or USSTAB= operands that are specified on the named TYPE=MXT REMOTE definition statement can supply different values than those coded on this REMOTE definition statement. If any or all of the callout operands are omitted from the named TYPE=MXT REMOTE definition, data is taken from the corresponding operand on this REMOTE definition.

Although the same TYPE=MXT REMOTE definition statement can be used for callout or callin SVCs, we recommend that you specify different TYPE=MXT REMOTEs for each type.

If a specified SVC3= suboperand is in error, it will be reported then bypassed.

If the number of SVC3= operand list entries is less than the ***vclmt*** value that is in effect, the unspecified entries will be ignored.

You may specify from 1 to ***vclmt*** SVC3= operand list entries.

Example:
```
GATE=GENERAL                  ; GATE FUNCTION REQUIRED
CONNECT=NO                    ; NO GATEFC
SUBADDR=YES                   ; SUBD SETS LLC
LLC3=(3,6)                    ; SUPPLY SUBD VALUES
SVC3=(2,                      ; ALLOW 2 LLC3 SVCS
      SVC3SPU1/3026001I,      ; SUPPLY LLC3 SVC SLU NAMES/ADDR
      SVC3SPU2/3026002I)
```

SVC3= Operand Run Time Processing

LLC3 application selection for SVCs is accomplished as follows:

1. For callin entries in the SVC3= operand list, HNAS will wait for the arrival of an XOT **inbound** Call Request packet. When an LLC3 call is received (as determined by CUD0 or a subaddress digit), a QLLC XID request is sent to the remote SPU to solicit its IDBLK/IDNUM values which are contained in the QLLC XID response.

2. When the QLLC XID response is received, the SVC3= operand is used for SPU selection. SPU allocation in the SVC3= operand list is performed left to right as follows:

If an SVC3= operand entry contains a decimal callin ***dteaddr*** value (designated with **I** as its connection identifier), it is compared against the ***calling*** DTE address from the original inbound Call Request packet. If a match occurs, the SPU is allocated to the VC. If no match occurs, the next SVC3= operand entry is examined. This form of SPU selection guarantees that the remote DTE will always access host applications using the same SPU resource.

If all SVC3= operand entries have been examined without a ***dteaddr*** match occurring, an SPU is dynamically allocated to the VC from the SPU pool based on IDBLK/IDNUM matching. If no SPU can be found with a matching IDBLK/IDNUM value, the call is cleared.

3. After SPU selection has been completed, the following packet exchanges take place between HNAS (acting as the SSCP) and the remote SPU.

```
HNAS                                                            SPU
-------------------------------------------------------------------

  QSM          -------------------------------->
               <--------------------------------  QUA
  ACTPU REQ    -------------------------------->
               <--------------------------------  ACTPU RSP
  ACTLU1 REQ   -------------------------------->
               <--------------------------------  ACTLU1 RSP
                                :
  ACTLUn REQ   -------------------------------->
               <--------------------------------  ACTLUn RSP
               <--------------------------------  FMDi REQ1
                                :
                    process FMD for logon text
                                :
  FMDi RSP1    -------------------------------->
```

In the sequence/response diagram above, an ACTLU is sent by HNAS for every SLU identified in the LUNAME= operand list on the selected SPU. Application (PLU) selection occurs when the first FMD PIU is received from an SLU based on, in the following order, (1) the DATA= suboperand values in the SYSL= operand list, (2) the LOGTAB= operand value and (3) the USSTAB= operand value via MCHSOL processing.

Note: If an INITSELF request is received, MCHSOL processing is activated unconditionally (bypassing SYSL=DATA= operand logic) using the application name text within the INITSELF PIU.

If application selection fails, a DACTLU is returned to the SLU. If application selection is successful, a REQSESS request is passed to the selected PLU on behalf of the SLU. If the selected host application is not active (REQSESS fails), a DACTLU is returned to the SLU. If the selected host application is active, it will send a BIND, SDT, etc. to the SLU. From this point on, the PLU is controlling the SLU and all exchanges take place between the PLU and SLU until the PLU issues an UNBIND. The HNAS SSCP function then resumes control of the SLU.

In this example, the SPU named SVC3SPU1 is allocated only if the ***calling*** DTE address is 30260001 while SVC3SPU2 is allocated only if the ***calling*** DTE address is 30260002. If a *calling* DTE address is any other value, an SPU is allocated dynamically from the SPU pool based on IDBLK/IDNUM matching. In this case, SVC3SPU1 or SVC3SPU2 may be allocated if the IDBLK/IDNUM values from the XID response match the values coded on their TYPE=SPU REMOTE definition statements. If no SPU can be matched, the call is cleared.

Note: If a session manager is employed for QLLC resources then **SESSLIM=YES** must be coded on the appropriate APPL statement in the HNAS VTAM Application Major Node File (AMNF) to ensure that BINDs are properly processed.

Primary Format

```
[SVC4={NONE|(vclmt,
            [{sluname[-{A|I}]|rmtname(4)||4||i}],...)}]
```

or, when you want to control SLU name generation

Alternate Format

```
[SVC4={NONE|(pfxlu,sfxst,vclmt)}]
```

(MCH|XTP client)
(changed for V2R3M0)
(changed for V2R4M0)

For the Primary Format

Specifies either **NONE** or an LLC4 Switched Virtual Circuit count (***vclmt***) followed by a mapping list that identifies, for each LLC4 SVC, an SLU name (***sluname***) that is assigned to the SVC. GATE=GENERAL and CONNECT=NO must also be specified.

The first SVC4= operand list entry specifies that LLC4 (GATE) SVCs are not used (**NONE**) or specifies an LLC4 SVC virtual circuit limit (***vclmt***).

If **NONE** is specified, LLC4 SVC support will not be provided for this MCH.

If a ***vclmt*** value is specified, LLC4 SVC support will be provided. In this case, the second SVC4= operand list entry is for the first LLC4 SLU, the third SVC4= operand list entry is for the second LLC4 SLU, and so on. HNAS SVC4= operand indices are independent of the SVC Logical Unit Numbers (LCNs) that are used by the router.

Once established, XTP SVC sessions communicate using XTP packets that carry an interface number to identify the MCH on the router and a circuit number to identify the LCN on the MCH. XOT SVC sessions communicate using XOT packets that carry an LCN. The MCH link is identified by the TCP/IP socket itself.

HNAS allocates and formats a Logical Unit Block (LUB) Virtual Circuit Block (VCB) for each SVC identified in the SVC4= operand.

The ***vclmt*** value you specify indicates how many LLC4 SVCs will be allowed for this MCH.

You may specify a ***vclmt*** value between 1 and 511. If a ***vclmt*** value is not specified or if the specified value is invalid, **an error message is generated if** GATE=GENERAL, SUB-ADDR=YES and LLC4=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,<28,...) are specified. Otherwise, **NONE** will be used.

Note: We suggest that you specify **NONE** for the SVC*i*= operands that you do not plan to use and a ***vclmt*** value for those that you do plan to use.

The ***sluname*** value you specify identifies a GATE terminal session SLU that can be accessed by multiple host applications as an LUT1 device. The ***sluname*** value can be any valid assembler language symbol.

If an ***sluname*** value is omitted (,), a **default SLU name** is generated from the first four (4) characters of the REMOTE name (***rmtname***), padded on the right with # or **£** characters (X'7B') as necessary, followed by the character '4' and the 3-digit hexadecimal SVC4= operand index value (plus one).

Note: Each of the specified or generated ***sluname*** values must also appear on an APPL statement in the HNAS VTAM Application Major Node File (AMNF).

You may specify **-{A|I}** after the SLU name to provide the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When the SLU initial state is active, the SLU is available for use in an LLC4 data session controlled by the CTCP. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to make the SLU available for GATE data sessions.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

If a specified SVC4= suboperand is in error, a default value will be substituted.

If the number of SVC4= operand list entries is less than the ***vclmt*** value that is in effect, default values will be assumed for the unspecified entries.

You may specify from 1 to ***vclmt*** SVC4= operand list entries.

```
Example: GATE=GENERAL                  ; GATE FUNCTION REQUIRED
         CONNECT=NO                    ; NO GATEFC
         SUBADDR=YES                   ; SUBD SETS LLC
         LLC4=(2,4,5)                  ; SUPPLY LLC4 SUBD VALUES
         SVC4=(2,                      ; ALLOW 2 LLC4 SVCS
         SVC4SLU1,SVC4SLU2)            ; SUPPLY LLC4 SVC SLU NAMES
         CUD0=(00,01,C0,NULL,02)       ; SUPPLY CUD0 VALUES
         CTCP=(0,0,1,2,0)              ; SUPPLY LUNAME INDICES
         LUNAME=(CTCPSLU0,             ; SUPPLY CTCP SLU NAMES
                 CTCPSLU1,
```

CTCPSLU2

For the Alternate Format

Specifies either **NONE** or an SLU name prefix value (***pfxlu***) followed by a suffix start value (***sfxst***) followed by an LLC4 Switched Virtual Circuit count (***vclmt***).

The primary and alternate specifications are both be accepted. The alternate specification is appropriate when no additional suboperands need to be associated with each SLU name. For the alternate specification:

The ***pfxlu*** value must be the first SVC4= suboperand and may be any valid assembler language symbol up to 7-characters in length starting with either an alpha character (A,B,C,...,Z) or an accepted special character (@, #, $ or %). This suboperand is REQUIRED to indicate that the alternate format is being used.

The ***sfxst*** value must be the second SVC4= suboperand and must be a hexadecimal number (without the framing characters X'') between 0 and F when the ***pfxlu*** length is 7, between 0 and FF when ***pfxlu*** length is 6, ..., between 0 and FFFFFFF when the ***pfxlu*** length is 1. If ***sfxst*** is omitted (,,), a default value of 0 will be used.

The ***vclmt*** value must be the third SVC4= suboperand and must be a decimal number between 0 and 511 (the SVC4= array size). This suboperand is REQUIRED. The SLU names that HNAS generates from the ***pfxlu***, ***sfxst*** and ***vclmt*** values will always be 8-characters in length with zero (0) pad characters added as required between the last ***pfxlu*** character and the suffix value.

Examples:

If SVC4=(XX,1,3) is specified, the generated SLU names would be XX000001, XX000002, XX000003.

If SVC4=(XXXXX,0,3) is specified, the generated SLU names would be XXXXX000, XXXXX001, XXXXX002.

SVC4= Operand Run Time Processing

LLC4 and CTCP selection is accomplished for SVCs as follows:

1. When an LLC4 call is received (as determined by CUD0 or a subaddress digit), the SVC4= operand is used for SLU selection. HNAS will allocate the first available SLU in the operand list. If no SLU is available, the call is cleared.

2. After SLU selection has been completed, the CUD0= and CTCP= operands are used for CTCP selection. A lookup is performed in the CUD0= operand list using the CUD0 byte from the inbound Call Request packet. The CUD0 byte is a binary number.

3. If no match is found, the call is cleared.

4. If a match occurs, the position of the matched CUD0 value in the CUD0= operand list is used as an index into the CTCP= operand list.

5. The CTCP= operand list entry is then used as an index into the LUNAME= operand list. If the selected CTCP SLU is not bound, the call is cleared.

6. In the example above, a CUD0 value 00, 01 or 02 results in the selection of CTCPSLU0, a CUD0 value C0 results in the selection of CTCPSLU1 and a NULL CUD0 value (no call user data at all) results in the selection of CTCPSLU2.

Primary Format

```
[SVC5={NONE|(vclmt,
           [{sluname[+gluname][-{A|I}]|rmtname(4)||5||i}]
           [/{X||idnum1|dteaddr1/<rmtname1>}
              [-{X||idnum2|dteaddr2/<rmtname2>}
                 [-{X||idnum3|dteaddr3/<rmtname3>}]]
                    ||{T|O|I}[{applid|0}]}]
           [/mxtname]
           [/cud],...)}]
```

<u>or, when you want to control SLU name generation and no additional parameters are required</u>

Alternate Format

```
[SVC5={NONE|(pfxlu,sfxst,vclmt)}]
```

(MCH|XTP client)
(new for V1R1M3)
(changed for V2R1M1)
(changed for V2R2M0)
(changed for V2R3M0)
(changed for V2R4M0)

<u>**For the Primary Format**</u>

Specifies either **NONE** or an LLC5 Switched Virtual Circuit count (***vclmt***) followed by a mapping list that identifies, for each LLC5 SVC, (1) an SLU name (***sluname***) that is assigned to the SVC and optionally and optionally a generic SLU name (***gluname***) for use by VTAM acquire operations (PLU callout), (2) a remote DTE identifier specified as a DTE address (***dteaddr***), a hex IDNUM value (**X*idnum***) or the name of a TYPE=MXT REMOTE definition statement (***<rmtname>***) whose DTEADDR= operand identifies the remote DTE, (3) a connection identifier ({**I**|**O**|**T**}) that determines the call setup procedure, (4) an application selection index (***applid***), (5) the name of a TYPE=MXT REMOTE definition statement (***mxtname***) that provides overriding operands and (6) call user data (***cud***) that will override any CUD= operand data and be placed in the outbound Call Request packet that HNAS creates when the SLU is bound (SLU is marked for callout (**O**|**T**)). GATE=NO|GENERAL and CONNECT=NO and PAD=INTEG|TRANSP must also be specified.

The first SVC5= operand list entry specifies that LLC5 (PAD) SVCs are not used (**NONE**) or specifies an LLC5 SVC virtual circuit limit (***vclmt***).

If **NONE** is specified, LLC5 SVC support will not be provided for this MCH.

If a ***vclmt*** value is specified, LLC5 SVC support will be provided. In this case, the second SVC5= operand list entry is for the first LLC5 SLU, the third SVC5= operand list entry is for the second LLC5 SLU, and so on. HNAS SVC5= operand indices are independent of the SVC Logical Unit Numbers (LCNs) that are used by the router.

Once established, XTP SVC sessions communicate using XTP packets that carry an interface number to identify the MCH on the router and a circuit number to identify the LCN on the MCH. XOT SVC sessions communicate using XOT packets that carry an LCN. The MCH link is identified by the TCP/IP socket itself.

HNAS allocates and formats a Logical Unit Block (LUB) and Virtual Circuit Block (VCB) for each SVC identified in the SVC5= operand.

The ***vclmt*** value you specify indicates how many LLC5 SVCs will be allowed for this MCH.

You may specify a ***vclmt*** value between 1 and 511. If a ***vclmt*** value is not specified or if the specified value is invalid, **an error message is generated if** GATE=NO and PAD=INTEG|TRANSP are specified **or if** GATE=GENERAL, SUBADDR=YES and LLC5=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,85,...) are specified. Otherwise, **NONE** will be used.

Note: We suggest that you specify **NONE** for the SVC*i*= operands that you do not plan to use and a ***vclmt*** value for those that you do plan to use.

The ***sluname*** value you specify identifies an PAD terminal session SLU that can be accessed by multiple host applications as an LUT1 device. The ***sluname*** value can be any valid assembler language symbol.

The ***gluname*** value (if specified) identifies a generic SLU name that will be passed to VTAM (SETLOGON OPTCD=GNAMEADD) when the ACB for ***sluname*** is opened. This allows PLUs to ACQIUIRE the HNAS resource (for callout operations) using ***gluname.*** Generic resources must also be defined in VTAM.

Note: ***sluname+gluname*** support was introduced into 240 as Enhancement APAR 2400084.

If an ***sluname*** value is omitted (/), a **default SLU name** is generated from the first four (4) characters of the REMOTE name (***rmtname***), padded on the right with **#** or **£** characters

(X'7B') as necessary, followed by the character '5' and the 3-digit hexadecimal SVC5= operand index value (plus one).

Note: Each of the specified or generated ***sluname*** values must also appear on an APPL statement in the HNAS VTAM Application Major Node File (AMNF).

You may specify **-{A|I}** after the SLU name to provide the initial SLU state (-A for active (online) or -I for idle (offline)). -A (active) is the default initial state. When the SLU initial state is active, the SLU is conditioned to accept or solicit LLC5 (PAD) SVC calls. When the SLU initial state is idle, the **VARY *sluname* ON** console command must be issued to activate the SLU for LLC5 SVC connections.

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

For all SVC5= operand entries, 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 be enclosed in non-equivalence symbols (<>) and must name a TYPE=MXT|SVC REMOTE definition statement. 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.

Inbound **_dteid_i** list processing:

When the **_dteid_i** list entries are identified as **I**nbound (the connection identifier is omitted or is specified as **I**) or **T**woway (the connection identifier is specified as **T**), the following rules are used to allocate an SLU using the **_dteid_i** values:

When a **_dteaddr_i** value is specified, it must match the ***calling*** DTE address in the incoming Call Request packet (for the number of digits specified by **_dteaddr_i**) in order for the associated SLU to be allocated. The **_dteaddr_i** represents the physical DTE at the X.25 network entry point which is the only DTE that will be allowed to use the SLU. If the **_dteaddr_i** value does not yield a match, the next **_dteid_i** list entry is examined.

When an **X*idnum*i** value is specified, it must match the Call User Data (starting at byte 1) in the incoming Call Request packet (for the number of bytes specified by **X*idnum*i**) in order for the associated SLU to be allocated. If the **X*idnum*i** value does not yield a match, the next **_dteid_i** list entry is examined. If the CUD0SELECTSLU OPTION is coded the **X*idnum*** compare starts at CUD0 (not CUD1).

Note: For NPSI emulation of CUD IDNUM data which is 5-digits in length, you would actually code 6-digits for **X*idnum*i**, the last being zero (0). For example, if the CUD for an incoming PAD Call Request packet is C5100020, the IDNUM portion is 10002. In the SVC5 entry, you would specify the **X*idnum*i** value as X100020 since the compare is done on CUD bytes 1-3.

Note: The same **X*idnum*i** value may be specified for different SLUs on different MCHs. The MCH is selected via RTEIN= mapping. Once the MCH is selected, the **X*idnum*i** value is then used to select the SLU.

When a **<*rmtname*i>** value is specified, it must identify a TYPE=MXT REMOTE definition statement whose DTEADDR=***dteaddr*i** value will be used to provide a DTE address for ***calling*** DTE address matching. If no ***dteaddr*i** value is given on the MXT or if the specified ***dteaddr*i** value does not yield a match, the next ***dteid*i** entry is examined.

If all ***dteid*i** list values have been tested without yielding a match, the SLU is not allocated and the call is cleared.

If no ***dteid*i** list is given, the SLU is allocated on a first come, first served basis.

If you specify a connection identifier character of **I|T** following the ***dteid*i** list, you may also specify a APPLNAME= operand index value (***applid***) following the **I|T**. You may specify an ***applid*** value between 0 and N'APPLNAME-1 (the APPLNAME= operand entry count minus one). If an ***applid*** value is specified, HNAS will use it to select a PLU name for a REQSESS request. This feature allows you to dedicate an SLU to a specific application (PLU) whose name appears in the APPLNAME= operand list. If an ***applid*** value is not specified, the SYSL= operand will be used to select an application. If the SYSL= operand is also not specified, a default ***applid*** value of 0 will be used to select the first APPLNAME= operand entry.

Outbound ***dteid*i** list processing:

When the ***dteid*i** list entries are identified as **O**utbound (the connection identifier is specified as **O**) or **T**woway (the connection identifier is specified as **T**), the following rules are used to establish a connection to the remote DTE using the ***dteid*i** values in an outbound Call Request packet:

When a ***dteaddr*i** value is specified, it is used as the ***called*** DTE address for the outbound call. The ***calling*** DTE address, facilities and call user data come from the DCEADDR=, FAC and CUD= operands on either the associated MXT (see ***mxtname*** without <> below) or the root MCH.

When a **<*rmtname*i>** value is specified, the DTEADDR=***dteaddr*i** and DCEADDR=***dceadd*i** operands on the named MXT are used to provide the ***called*** and ***calling*** DTE addresses, respectively, for the outbound call. The FAC= and CUD= operands on the same MXT provide the facilities and call user data for the outbound call.

Note: When an MXT or SVC is specified as a *rmtname*i in place of a *dteaddr*i value, it becomes the REMOTE that is associated with the SLU for the duration of the LU/VC session. In this case, it will be the REMOTE that provides the *called* DTE address (DTEADDR= operand), *calling* DTE address (DCEADDR= operand), facilities data (FAC= operand) and call user data (CUD= operand). These four operands provide all the information necessary to create a unique Call Request packet.

For PAD calls that require PADPARM= values to override those on the root MCH, an MXT rather than an SVC should be used in place of a *dteaddr*i value. This is because the SVC REMOTE does not support the PADPARM= operand.

If PADPARM= values are not required, we recommend using an SVC rather than an MXT for call processing. The SVC control block uses less memory than the MXT. The SVC control block is X'178' (376) bytes in length while the MXT control block is X'480' (1152) bytes in length.

Note: <*rmtname*i> support was introduced into 240 as Enhancement APAR 2400003.

The purpose in providing multiple callout DTE addresses is to allow HNAS to retry failed outbound call requests. Make sure that all ***dteaddr*i** values pass the DTE address filter that you have specified for the RTEOUT operand on the HOME TYPE=XOT LOCAL definition statement. This will permit HNAS to allocate a TCP socket for the XOT connection. Each ***dteaddr*i** will be used, in turn, as the ***called*** DTE address in the outbound Call Request packet that HNAS creates when the SLU is bound if the call is cleared by the remote DTE.

For example, if 3 DTE addresses are specified for an SLU, ***dteaddr1*** will be used when the SLU is initially bound by the PLU. If HNAS receives a Clear Request packet as the response to its Call Request packet Call Request times out (nothing received), a new call will be attempted using ***dteaddr2***. If this call also fails, a new call will be attempted using ***dteaddr3***. If this call also fails, the PLU will be notified via an UNBIND request. If a Call Accepted packet is received as a response to any of the Call Request packets, normal processing will continue.

Example: `SVC5=(...,MCH10020/X900060-20360009-<MTX0>T2/MXT1/C500000,...`

The ***mxtname*** value you specify after the DTE address list identifies an MCH extension (MXT) that supplies overriding operands like CUD=, DCEADDR= and/or FAC= for callout and/or LOGTAB= and USSTAB= for callin. The name you specify for ***mxtname*** must appear in the name field of a TYPE=MXT REMOTE definition statement that appears elsewhere in the CDF. The ***mxtname*** value can be any valid assembler language symbol.

The CUD=, DCEADDR=, FAC=, LOGTAB=, PADPARM= and/or USSTAB= operands that are specified on the named TYPE=MXT REMOTE definition statement can supply different values than those coded on this REMOTE definition statement. If any or all of the callout operands are omitted from the named TYPE=MXT REMOTE definition, data is taken from the corresponding operand on this REMOTE definition.

Although the same TYPE=MXT REMOTE definition statement can be used for callout or callin SVCs, we recommend that you specify different TYPE=MXT REMOTEs for each type.

For SVC5= operand entries identified as **O**utbound (the connection identifier is specified as **O**) or **T**woway (the connection identifier is specified as **T**), you may specify a ***cud*** value that will be placed in the Call User Data field of the outgoing Call Request packet that HNAS creates. This ***cud*** value overrides the CUD= operand value from this REMOTE definition statement and from the TYPE=MXT REMOTE definition statement that is identified by ***mxtname***. If the CUD= operand is also omitted, a default value of 01000000 will be used.

If specified, the SVC5= ***cud*** value must be **exactly 4 bytes** (8 digit pairs) in length. If you require more or less than 4 bytes of call user data, you will have to code the CUD= operand on the TYPE=MXT REMOTE definition statement identified by ***mxtname*** or on this TYPE=MCH REMOTE definition statement.

Call Request Parameter Hierarchy Matrix

PARAMETER	When *dteid*i=<*rmtname*i>	When *dteid*i=*dteaddr*i
called **DTE address**	**DTEADDR= from <*rmtname*i>**	***dteaddr*i**
calling **DTE address**	**DCEADDR= from <*rmtname*i>**	**1) DCEADDR= from *mxtname* if defined** **2) DCEADDR= from root MCH**
facilities data	**FAC= from <*rmtname*i>**	**1) FAC= from *mxtname* if defined** **2) FAC= from root MCH**
call user data	**CUD= from <*rmtname*i>**	**1) *cud* from SVC0 operand if defined** **2) CUD= from *mxtname* if defined** **3) CUD= from root MCH**

Note: If the operand at the bottom of the hierarchy list for a particular Call Request parameter is also undefined (null), the corresponding Call Request field will contain zeros.

If a specified SVC5= suboperand is in error, a default value will be substituted.

If the number of SVC5= operand mapping list entries is less than the ***vclmt*** value that is in effect, default values will be assumed for the unspecified entries.

You may specify from 1 to ***vclmt*** SVC5= operand list entries.

```
Example: GATE=GENERAL                     ; GATE FUNCTION REQUIRED
         CONNECT=NO                       ; NO GATEFC
         PAD=TRANSP                       ; XPAD SUPPORT REQUIRED
         SUBADDR=YES                      ; SUBD SETS LLC
         LLC5=(7,8)                       ; SUPPLY LLC5 SUBD VALUES
         SVC5=(2,                         ; ALLOW 2 LLC5 SVCS
         SVC5SLU1/3026051I,               ; SUPPLY LLC5 SVC SLU NAMES/ADDR
         SVC5SLU2/3026051O)
         SYSL=(CUD4=C5/0)                 ; SUPPLY APPL MAPPING
         APPLNAME=(USER)                  ; IDENTIFY NON-CTCP APPS
```

For the Alternate Format

Specifies either **NONE** or an SLU name prefix value (***pfxlu***) followed by a suffix start value (***sfxst***) followed by an LLC5 Switched Virtual Circuit count (***vclmt***).

The primary and alternate specifications are both be accepted. The alternate specification is appropriate when no additional suboperands need to be associated with each SLU name. For the alternate specification:

The ***pfxlu*** value must be the first SVC5= suboperand and may be any valid assembler language symbol up to 7-characters in length starting with either an alpha character (A,B,C,...,Z) or an accepted special character (@, #, $ or %). This suboperand is REQUIRED to indicate that the alternate format is being used.

The ***sfxst*** value must be the second SVC5= suboperand and must be a hexadecimal number (without the framing characters X'') between 0 and F when the ***pfxlu*** length is 7, between 0 and FF when ***pfxlu*** length is 6, ..., between 0 and FFFFFFF when the ***pfxlu*** length is 1. If ***sfxst*** is omitted (,,), a default value of 0 will be used.

The ***vclmt*** value must be the third SVC5= suboperand and must be a decimal number between 0 and 511 (the SVC5= array size). This suboperand is REQUIRED. The SLU names that HNAS generates from the ***pfxlu***, ***sfxst*** and ***vclmt*** values will always be 8-characters in length with zero (0) pad characters added as required between the last ***pfxlu*** character and the suffix value.

Examples:

If SVC5=(XX,1,3) is specified, the generated SLU names would be XX000001, XX000002, XX000003.

If SVC5=(XXXXX,0,3) is specified, the generated SLU names would be XXXXX000, XXXXX001, XXXXX002.

SVC5= Operand Run Time Processing

Application (PLU) selection for LLC5 SVCs is accomplished as follows:

1. For callout entries in the SVC5= operand list, HNAS will set LLC5 and generate an XTP or XOT **outbound** Call Request packet when the SVC5 SLU is acquired (bound). No additional system selection is required because the SVC5 SLU is ‘owned’ by the binding host application.

2. For callin entries in the SVC5= operand list, HNAS will wait for the arrival of an **inbound** Call Request packet. When an LLC5 call is received (as determined by CUD0 or a subaddress digit), the SVC5= operand is used for SLU selection. SLU allocation is a two pass operation performed as follows:

   Pass 1 (scan entries with idnum or DTE address values)
   If an SVC5= operand entry contains a hex **X*idnum*** value, it is compared against the IDNUM value from the **CUD** field of the inbound Call Request packet (byte 1-n, where n is the number of bytes you code for ***idnum***). If a match occurs, the SLU is allocated to the VC. If no match occurs, the next SVC5= operand entry is examined. If the CUD0SELECTSLU OPTION is coded the **X*idnum*** compare starts at CUD0 (not CUD1).

   If an SVC5= operand entry contains a decimal callin ***dteaddr*** value (designated with **I** as its connection identifier), it is compared against the ***calling*** DTE address from the inbound Call Request packet (**for the number of digits specified by *dteaddr***). If a match occurs, the SLU is allocated to the VC. If no match occurs, the next SVC5= operand entry is examined. This form of SLU selection guarantees that the remote DTE will always access host applications using the same SLU resource.

   Pass 2 (scan entries with no idnum or DTE address values)
   If no match is found in the SVC5= entries with idnum or DTE address values then the

remaining SVC5= entries are searched for an available SLU. If one is found it is allocated to the VC.

If all SVC5= operand entries have been examined and no SLU can be allocated, the call is cleared.

3. After SLU selection has been completed, the SYSL= operand is used for application (PLU) selection. The SUBD= and/or CUD*i*= suboperands of the SYSL= operand specify mapping values and indices into the APPLNAME= operand list.

   If the SYSL= operand is omitted or if no match occurs in the SYSL= operand list, the first APPLNAME= operand entry will be used as the default application. If the selected host application is not active (REQSESS rejected) or if the Console Subsystem is selected and it is unavailable, the call is cleared.

   If an APPLNAME= operand index (***applid***) is coded for the SLU after the **X*idnum*** or ***dteaddr*** delimiter character **I**, this index used to locate an application in the APPLNANE= operand list. Since no user input is required for application selection, the SLU is effectively dedicated to one application (PLU).

4. In the example above, a CUD4 value C5 results in the selection of USER.

```
[SYSL=({SUBD=subdval/index|CUD0=cudval/applid|
        CUD1=cudval/applid|CUD2=cudval/applid|
        CUD3=cudval/applid|CUD4=cudval/applid|
        CUD5=cudval/applid|CUD6=cudval/applid|
        CUD7=cudval/applid|CUD8=cudval/applid|
        CUD9=cudval/applid|NULL/applid|
        DATA=textstring/applid},...)]
                                        (SPU|MCH|XTP client)
                                        (changed for V2R2M0)
```

For a TYPE=MCH|XTP REMOTE definition statement, the SYSL= operand specifies a lookup list that is used to map the SUBD and/or CUD*i* bytes (i = 0 to 9) from XTP or XOT **inbound** Call Request packets to APPLNAME= operand entries for non-GATE SVCs. Together, these operands provide a ‘system select’ capability for PCNE (LLC0) and PAD (LLC5) sessions. The SYSL= operand is **optional** but the APPLNAME= operand is **required** for PCNE and PAD sessions. If the SYSL= operand is omitted, the first (or only) APPLNAME= operand entry is used as the default application.

The SYSL= operand specifies a list of values that when matched by the selected byte from an XTP or XOT inbound Call Request packet, results in the selection of a particular host application.

For the **SUBD=** suboperand, you may specify a ***subdval*** between 0 and 99. When a one (1) digit **SUBD** value is specified (e.g., 9), only the last digit of the ***called*** DTE address field is tested for a subaddress match. When a two (2) **SUBD** value is specified (e.g., 09), the last two (2) digits of the ***called*** DTE address is tested for a subaddress match. Since the entire ***called*** DTE address is given in packed decimal format, **SUBD** values can range from 0 to 99.

For the **CUD*i*=** suboperands, you may specify a ***cudval*** value as a hexadecimal number between 0 and FF without the framing characters “X” or as **NULL** (the NULL value can only be used once). The **NULL** value provides host application selection when the XTP or XOT **inbound** Call Request packet does not contain any Call User Data.

You may specify ***applid*** values between 0 and the APPLNAME= operand entry count minus one (N’APPLNAME-1).

If a ***subdval***, ***cudval*** or ***applid*** value is not specified or if the specified value is invalid, the suboperand will be ignored.

You may specify from 1 to 255 SYSL= operand list entries.

An Example for SYSL= operand processing can be found on page 4-77 of this document.

<u>For a TYPE=SPU REMOTE definition statement</u>, the SYSL= operand specifies a lookup list that is used to map terminal input from native SNA SLUs to APPLNAME= operand entries. Together, these operands provide a ‘system select’ capability for QLLC SLU sessions.

For the **DATA=** suboperand, you may specify a ***textstring*** value up to 10 characters in length. The value you specify is compared against data from the initial FMD PIU received from an SLU after it has been activated via an ACTLU request.

Note: If the SYSL= operand is not specified or if no match is made for any of its DATA= suboperands, normal MCHSOL processing will be used for application selection via the LOGTAB= and/or USSTAB= operands.

Note: If an INITSELF request is received, MCHSOL processing is unconditionally activated (bypassing SYSL=DATA= operand logic) using the application name text within the INITSELF PIU.

You may specify ***applid*** values between 0 and the APPLNAME= operand entry count minus one (N’APPLNAME-1).

If a ***textstring*** or ***applid*** value is not specified or if the specified value is invalid, the suboperand will be ignored.

You may specify from 1 to 69 SYSL= operand list entries.

[TAP={*seconds*|<u>0</u>}]

(XOT|XTP client)
(changed for V2R1M0)
(changed for V2R2M0)

Specifies a shoulder TAP timeout value for this HNAS client component. HNAS shoulder tap (Keep Alive) processing is different but can be used in conjunction with TCP/IP Keep Alive processing. The former uses XTP or XOT protocol packets for Keep Alive exchanges to ensure that the router transport protocol is still functioning. The latter uses standard TCP/IP SEQNO/ACK (or 1 one-byte 'garbage') packet exchanges to ensure that the socket connection is still viable and is invoked by the **KEEPALIVEOPTIONS** or **INTERVAL** statement in the **TCPIP PROFILE** file.

For XOT (Cisco routers), HNAS will transmit a special Call Request packet containing data you provide via the DCEADDR=, DTEADDR=, FAC= and CUD= operands at the interval specified by the TAP=***seconds*** value. The Call Request packet will contain the following data:

Packet Type: 0B (Call Request).
Calling DTE Address: DCEADDR= operand value or null if omitted.
Called DTE Address: DTEADDR= operand value or null if omitted.
Facilities Data: FAC= operand value or null if omitted.
Call User Data: CUD= operand value or X'01000000',C'HNASTAP' if omitted.

Note: DCEADDR=, DTEADDR=, FAC= and CUD= operand support was added for TYPE=XOT REMOTE definition statements starting with the V2R3M0 release of HNAS. These operands are only allowed when tapping is enabled (TAP=*seconds* specified). These operands allow XOT shoulder tap processing to include *calling*/*called* DTE addresses, facilities and call user data parameters within the XOT tapping Call Request packet. This means that you have far greater control over **routing** and **filtering** of the Call Request packet within the router. XOT tapping calls can now be **selectively cleared** (thus providing the necessary tapping response) using the Cisco '**x25 route *dteaddr* clear**' configuration option.

Note: If the DTEADDR= operand is omitted or if the specified DTE address is not defined in a router routing statement, the router will return a Clear Request to HNAS. This is considered a test of HNAS-to-router connectivity. If, however, the DTEADDR= operand specifies a DTE address that is defined in a router X.25 link routing statement, the response returned to HNAS comes from the addressed remote DTE. This is considered a test of HNAS-to-remote (or end-to-end) connectivity which also provides an indication that both the router and the physical X.25 serial link are operational. If you do not wish to have tapping Call Request packets sent out across a physical X.25 link, we suggest and recommend that you specify a DTEADDR= value that matches the *dteaddr* in a Cisco '**x25 route *dteaddr* clear**' configuration option. This will cause a Clear Request to be returned by the router and prevent the Call Request from being transmitted across a physical X.25 link. Should you wish to test end-to-end connectivity, you can use HNAS PING XOT console command.

Note: A Call Request packet was chosen as the XOT tap request because a special protocol level Keep Alive packet was not defined by the XOT specification. The use of the Call Request packet as the shoulder tap request not only serves to test router connectivity, it also tests the viability of the XOT routing within the router. The shoulder tap will fail even if the router is connected and can be PINGed if XOT routing is not enabled in the router.

A response (normally a Clear Request packet) must be returned to HNAS within an interval equal to the ***seconds*** value. The tapping process is considered successful if *any* response is returned to the tapping Call Request packet. Because each virtual circuit connection to the router requires its own TCP socket, a special (extra) TCP socket is used just for shoulder tapping. This socket operates independently from all others.

The shoulder tap request from HNAS informs the router that its connection to HNAS is still operational. The shoulder tap response from the router informs HNAS that its connection to the router is still operational. Although a specific shoulder tap response is expected from the router, any form of input will satisfy the HNAS shoulder tap response criteria.

If no input is received within the shoulder tap response interval, a **NAS2501W** message will be issued.

If two (2) consecutive shoulder tap failures occur (router contact lost), the router is taken offline (made temporarily idle), <u>all connections to the router are closed</u> and a **NAS2502E** message is issued. Further non-tapping Call Request packets will not be sent to this router as long as it remains offline.

Note: If you specify **OPTIONS=(...,NOCLOSEONTAPFAILURE,...)** for a TYPE=XOT|XTP REMOTE, HNAS <u>will not close</u> active sockets and their VC/LU components when a router contact loss condition is detected, that is, when 2 consecutive TAP (Keep Alive) failures occur. The sessions will remain active even though the router is marked as offline. This means that a customer who would like to monitor the TAP process (MONTAP) and leave the connections active even when it appears that the router is down can do so.

HNAS will continue to perform the shoulder tap function after contact to the router has been lost in an effort to re-establish communication. In this mode, the shoulder tap interval is adjusted (lengthened) to prevent flailing. The new interval is computed using the ***seconds*** value multiplied by the shoulder tap retry count (currently fixed at 2).

When contact to the router is reacquired, a **NAS2503W** message will be issued.

Note: For a description of the NAS2501W, NAS2502E and NAS2503W messages, please refer to the HNAS Messages and Codes document.

You may specify a ***seconds*** value between 0 and 4095. A value of 0 prevents HNAS from performing the shoulder tap function (and prevents the use of the DCEADDR=, DTEADDR=, FAC= and CUD= operands). Inhibiting HNAS shoulder tapping may be desired if you have other means of monitoring your router connections and detecting failures. If a ***seconds*** value is not specified or if the specified value is invalid, a default value of **0** will be used.

While values between 1 and 4095 will activate HNAS shoulder tapping, we do not recommend using values less than 10 nor greater than 180. We recommend a value between 30 and 120.

Note: A ***seconds*** value less than 10 can induce non-productive CPU overhead. Values greater that 180 (for Cisco routers) can cause ‘XOT VC timeout’ messages to be displayed at the router. This can occur because Cisco routers expect data from HNAS no later than 200 seconds after the TCP socket has been established. Conversely, HNAS will not send a tap request until its tap timer expires and this will only occur after the TCP socket has been established. Since both socket end points require an active socket connection before their data timers are started, you should set the HNAS tap value less than ‘XOT VC timeout’ value of 200 so that HNAS will tap the router before the router's timeout expires. This will inhibit the 'XOT VC timeout' message. The Cisco router sends a TCP FIN request to the TCPIP stack when it writes the 'XOT VC timeout' message. Therefore, in spite of the message, this event actually satisfies the HNAS shoulder tap criteria since it is perceived by HNAS as input from the router.

Note: When multiple REMOTE definition statements are specified for the same router (the same IPADDR= operand value with separate VCLMT= operand values for XOT), the TAP= operand value from the **first** REMOTE definition statement for the router will be used.

Note: Shoulder tapping (Keep Alive) processing is inhibited for TYPE=XOT REMOTEs when dynamic IP address assignment (IPADDR=DYNAMIC) is in affect for a router.

Note: For HNAS releases prior to V2R2M0, a Clear Request packet rather than a Call Request packet was used as the Keep Alive request. We have observed that some Cisco router environments (i.e., IOS level) **do not respond** to an XOT Clear Request packet with an XOT Clear Confirmation packet which prevented our Keep Alive simulation logic from operating properly. For this reason we had changed the default TAP=10 value to TAP=0 (none). Since HNAS now uses a Call Request packet as the XOT Keep Alive request, we expect all routers to respond regardless of the IOS level. The Cisco 'debug x25 event' command can be enabled to monitor the Keep Alive exchange.

Note: For HNAS release V2R2M0, OPTIONS=(TAPWITHCLR) was added for TYPE=XOT REMOTE definition statements to condition HNAS to use a Clear Request packet rather than a Call Request packet for the tap Keep Alive request (remembering that a Call Request is the new default Keep Alive request as of V2R2M0). The means that users who prefer to use a Clear Request packet for tapping can continue to do so.

Note: Trace entries for the tapping process are *not logged* when TRCVC or TRCMCH OCR are enabled. To trace the tapping process, you must use the MON TAP console command.

For XTP (IBM routers), HNAS will transmit a Keep Alive packet to the IBM router at the interval specified by the ***seconds*** value. A Keep Alive response must be returned to HNAS within an interval twice the ***seconds*** value. Because a single TCP socket is used for all virtual circuit connections to the router, the shoulder tap exchange is interleaved with all other traffic between HNAS and the router. For proper control of the shoulder tap process, the ***seconds*** value should match the Keep Alive timeout value that is configured in the router.

Note: When multiple REMOTE definition statements are specified for the same router (e.g., the same IPADDR= and PORT= operand values but different IFNUM= operand values for XTP), the TAP= operand value from the **first** REMOTE definition statement for the router will be used.

[TRAN={<u>NO</u>|USER|EVEN|ODD|MARK|SPACE|
NPSIEVEN|NPSIODD|NPSIMARK|NPSISPACE}]

(MCH|XTP client)
(changed for V2R3M0)

Specifies whether PAD data is to be translated from ASCII to EBCDIC on input and from EBCDIC to ASCII on output. In addition, indicates whether the ASCII parity bit will participate in the translation process. PAD=INTEG|TRANSP must also be specified.

<u>**NO**</u> specifies that no translation is to be performed. Data is passed transparently between the remote DTE and host application.

USER specifies that translation is to be performed using user supplied input and output translate tables.

EVEN specifies that translation is to be performed and that even parity ASCII is forced before input translation and after output translation. HNAS translate tables are used.

ODD specifies that translation is to be performed and that odd parity ASCII is forced before input translation and after output translation. HNAS translate tables are used.

MARK specifies that translation is to be performed and that the ASCII parity bit is set to one before input translation and after output translation. HNAS translate tables are used.

SPACE specifies that translation is to be performed and that the ASCII parity bit is set to zero before input translation and after output translation. HNAS translate tables are used.

NPSIEVEN, **NPSIODD**, **NPSIMARK** and **NPSISPACE** are treated the same as EVEN, ODD, MARK and SPACE, respectively, except that NPSI translate tables are used instead of HNAS translate tables.

Note: The TRAN=USER parameter is provided as an indicator of future support.

Note: For TRAN=EVEN|ODD|MARK|SPACE, standard HNAS translate tables are used. These tables are equivalent to the tables used for Comm-Pro's 37XX FEP Network Access Support (FNAS) program product.

Note: For TRAN=NPSIEVEN|NPSIODD|NPSIMARK|NPSISPACE, standard NPSI translate tables are used. These tables are equivalent to the tables used for IBM's 37XX FEP Network Package Switching Interface (NPSI) program product. We have found that for host applications like TPE, NPSI translate tables are more appropriate and are even required to eliminate communications problems.

If the TRAN operand is not specified or if the specified value is invalid, a default value of **NO** will be used.

Note: HNAS is distributed with a collection of PAD translate tables (both HNAS and NPSI versions) that are used to process the TRAN= operand for a TYPE=XTP|MCH REMOTE definition statement. These tables are contained in module **MCHTTBLS** which is a member of the **HNASOBJ** library. The standard HNAS translate tables work well for most host applications. However, for some host applications, the standard HNAS translate tables can cause problems which prevent coherent communication. This problem was first encountered for CICS when accessed by **TPE terminals** (Terminal de Paiement Electronique in French or Electronic Payment Terminal in English). This is due to the way some special characters are mapped.

For this reason HNAS now provides support for standard NPSI translate tables. We have found that this collection of NPSI translate tables eliminates problems with CICS when accessed by TPE terminals and may offer a solution for other host applications and other remote terminals.

The following illustrates some of the differences between the mappings provided by HNAS and NPSI translate tables.

```
               HNAS           NPSI
ASCII          EBCDIC         EBCDIC

X'21' -> !     X'5A' -> !     X'4F' -> |
X'5B' -> [     X'AD' -> [     X'4A' -> cent sign
X'5D' -> ]     X'BD' -> ]     X'5A' -> !
```

Note: For those customers who wish to create their own translate table module, source code is provided for the MCHTTBLS module in the HNASMAC library. If you wish to change the way characters are translated for PAD sessions, copy the MCHTTBLS source member from HNASMAC to HNASMACX then edit the copied member in HNASMACX. When complete, assemble the edited source member and stow the assembled object code in the HNASOBJX library as member MCHTTBLS. Relinking HNAS will then pick up your modified translate tables. For sample assembler JCL, please see the NASASM inline PROC in the HNASMNT job which is created from the original HNAS install JCL.

[TYPE={XTP|XOT|MCH|MXT|SPU1|SPU2|SPU|DFL|DFS|DFX|DMY|SVC}]
(new for V1R1M2)
(changed for V2R2M0)
(changed for V2R4M0)

Specifies the type of this HNAS client component.

XTP specifies that HNAS will provide XTP client services for this REMOTE definition.

XOT specifies that HNAS will provide XOT client services for this REMOTE definition. The name of this REMOTE definition statement is referenced as a ***rmtname*** value within the RTEOUT= operand of a TYPE=XOT LOCAL definition statement.

MCH specifies that HNAS will provide logical MCH services for this REMOTE definition. The name of this REMOTE definition statement is referenced as an ***mchname*** value within the RTEIN= operand of a TYPE=XOT LOCAL definition statement.

MXT specifies that HNAS will use this MCH extension to supply additional data for callin or callout support. The name of this REMOTE definition statement is referenced as an ***mxt-name*** value within the SVC0=, SVC3= or SVC5= operand of a TYPE=MCH|XTP REMOTE definition statement. MCH extensions may specify CUD=, DCEADDR=, FAC=, LOGTAB=, PADPARM= and USSTAB= operands only. The CUD=, DCEADDR= and FAC= operands are only used for callout. The LOGTAB= and USSTAB= operands are only used for callin. The PADPARM= operand is only used for LLC5 support.

SPU1 specifies that HNAS will provide SPU Type 1 (**PUT1**) client services for this REMOTE definition. The name of this REMOTE definition statement is referenced as an ***spuname*** value within the SVC3= operand of a TYPE=MCH REMOTE definition statement.

Note: SPU1 is used for QLLC support only.

SPU2 specifies that HNAS will provide SPU Type 2 (**PUT2**) client services for this REMOTE definition. The name of this REMOTE definition statement is referenced as an ***spuname*** value within the SVC3= operand of a TYPE=MCH REMOTE definition statement.

Note: SPU2 is used for QLLC support only.

SPU is treated the same as **SPU2**.

DFL specifies that HNAS will use this REMOTE to service Datafono leased SLU allocation requests for Datafono SVCs. The name of this REMOTE definition statement is referenced as a ***dflname*** value within the DFLNAME= operand of a TYPE=MCH REMOTE definition statement. For additional information on TYPE=DFL REMOTE definition statements, see Datafono TYPE=DFL REMOTE Definition description on page 4-166.

DFS specifies that HNAS will use this REMOTE to service Datafono switched SLU allocation requests for Datafono SVCs. The TYPE=DFS REMOTE definition statement is used to create a pool of switched resources for Datafono support. The resources are available for non-XID=NO calls received on any MCH. For additional information on TYPE=DFS REMOTE definition statements, see Datafono TYPE=DFS REMOTE Definition description on page 4-167.

DFX specifies that HNAS will use this REMOTE to provide Datafono session establishment services Datafono SVCs. The name of this REMOTE definition statement is referenced as a ***dfxname*** value within the DFXNAME= operand of a TYPE=MCH REMOTE definition statement. For additional information on TYPE=DFX REMOTE definition statements, see Datafono TYPE=DFX REMOTE Definition description on page 4-168.

DMY specifies that HNAS will use this REMOTE to provide TCPIP socket and X.25 Call Request packet parameters for the PING console command. The name of this REMOTE definition statement is referenced by the ***dmyname*** parameter in the PING command. For additional information on TYPE=DMY REMOTE definition statements, see PING TYPE=DMY REMOTE Definition description on page 4-161.

SVC specifies that HNAS will use this REMOTE to provide X.25 callin and/or callout parameters for LLC0 (PCNE) and/or LLC5 (PAD) SLU defined in the SVC0= or SVC5= operand of an

MCH specified somewhere in the CDF. The name of this REMOTE definition statement is referenced by the ***rmtnamei*** suboperand of the *dteidi* list of a SVC0= or SVC5= operand. For additional information on TYPE=SVC REMOTE definition statements, see SVC callout and callout processing description on page 4-130.

If the TYPE= operand is not specified or if the specified value is invalid, a default value of **XTP** will be used.

Note: HNAS can provide XTP and XOT services simultaneously. Multiple definitions for each type of client are permitted.

[USSTAB={*name*|BUILD}]

(SPU|MCH|MXT|XTP client)
(new for V1R1M4)
(changed for V2R2M0)

Specifies the name of a USS table that will be used for PCNE and/or PAD application LOGON interpret processing by the MCHSOL routine. MCHSOL processing is activated when MCHSOL is selected from the APPLNAME= operand. For more information on MCH-SOL services, see the description of the APPLNAME= operand on page 4-74 of this document.

Note: If both **LOGTAB=** and **USSTAB=** are specified, **LOGTAB=** is processed first by MCHSOL.

The ***name*** value you specify can be any valid assembler language symbol.

If a ***name*** value is not specified or if the specified value is invalid or if the specified value is not found in the library identified by the **VTAMLIB DD** statement, the default value will be taken from the **USSTAB=** operand on the **BUILD** definition statement.

[VCLMT={*count*|1|
+PVC+SVC0+SVC3+SVC4+SVC5|
+LUNAME(*slucnt*)}]

(DFS|DFL|MCH|XOT|XTP client)
(changed for V2R2M0)
(changed for V2R4M0)

For a TYPE=MCH|XTP REMOTE definition statement, specifies the number of entries to be reserved in the Address Vector Table (AVT) for the MCH. The AVT contains pointers to Virtual Circuit Blocks (VCBs) that are used to manage XTP or XOT sessions.

You may specify a ***count*** value between 1 and 65535.

If a ***count*** value is not specified or if the specified value is invalid, HNAS will use the sum of the ***vclmt*** values specified for the PVC=, SVC0=, SVC3=, SVC4= and SVC5= operands for a non-GATEFC MCH or the sum of the ***slucnt*** values from the LUNAME= operand for a

GATEFC MCH. The computed or specified VCLMT= operand value should match the total number of virtual circuits (PVCs and SVCs) that are defined for the physical X.25 MCH on the router.

For a TYPE=XOT REMOTE definition statement, specifies the number of TCP sockets that will be reserved for XOT client access. A single TYPE=XOT REMOTE definition statement can be used to specify multiple XOT clients. The VCLMT= operand value determines how many XOT client sockets will be generated.

You may specify a ***count*** value between 1 and 65535.

If a ***count*** value is not specified or if the specified value is invalid, a default value of **1** will be used.

Note: When a LOCAL is HOME to multiple TYPE=XOT REMOTEs whose total socket count (sum of the VCLMT= operand values) is greater than 2000, you must specify a SOCLMT= value larger than the VCLMT= sum. In addition to the REMOTE VCLMT= values, a TAP socket is created for each unique REMOTE IPADDR= address (this does not apply to IPADDR=DYNAMIC). The SOCLMT= should always be rounded to the next 2K boundary.

For example, if 3 TYPE=XOT REMOTEs with unique IPADDR= values have VCLMT= values of 2000, 1000 and 1000, the total remote sockets would be 4003 which includes the 3 TAP sockets. So for these 3 TYPE=XOT REMOTEs, the HOME LOCAL should specify SOCLMT=6000.

For a TYPE=DFL|DFS REMOTE definition statement, specifies the number of SLUs that will be available in the leased (DFL) or switched (DFS) SLU pools for Datafono terminal sessions.

You may specify a ***count*** value equal to the number of entries in the LUNAME= operand (***slucnt***).

If a ***count*** value is not specified or if the specified value is invalid, HNAS will use the ***slucnt*** value for the LUNAME= operand.

PING REMOTE TYPE=DMY Definition Statement

(new for V2R4M0)

The TYPE=DMY REMOTE definition statement is used to supply parameters to expand the special HNAS **PING *dmyname*** console command. You may code any number of TYPE=DMY remotes.

There are no restrictions on user assigned TYPE=DMY REMOTE names as long as the names are not defined for other remote types. Please refer to ***rmtname*** description below for additional information.

rmtname **REMOTE TYPE=DMY operands**	**Comments**	**CC/ VRM**
TYPE=DMY		N/240
[CUD={NONE|[(]*xx...xx*[)]}]	Recommend: H N A S P I N G CUD=01000000C8D5C1E2D7C9D5C7	
[DCEADDR={NONE|*dceaddr*}]	Calling X.25 DTE address	
[DTEADDR={NONE|*dteaddr*}]	Called X.25 DTE address	
[FAC={NONE|[(]*xx...xx*[)]}]	Recommend: FAC=0101420707430202	
[IPADDR=*a.b.c.d*]	Router IP address	
[PORT={1998| <------------- XOT 3065}] <------------- XTP	Router Port address	
[PROTOCOL={XOT|XTP}]	Call Request Transport Protocol	

Note: Common remote operands not provided under this TYPE=DMY remote type are located under the under the global REMOTE on page 4-69.

rmtname

Provides a name for the REMOTE definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF.

This remote is currently only referenced by the PING *rmtname* console command.

For other remote types ***rmtname*** will also appear in the RTEIN= and/or RTEOUT= operand on one or more LOCAL definition statements or in various operands on other REMOTE definition statements.

[PROTOCOL={XOT/XTP]

(restored for V2R4M0)

Specifies the router's protocol type. TCP port number for this HNAS server component.

You may specify **XOT** for Cisco routers or **XTP** for IBM routers. TCP port number values of 1998 (XOT) and 3065 (XTP) will be used when the socket is open for the HNAS PING request.

Datafono REMOTE TYPE=MCH Definition Statement

(changed for V2R4M0)

New operands for the TYPE=MCH REMOTE statement define the mechanisms used to create a Datafono session and to specify the session attributes. Datafono sessions support point of sale equipment used in Spain. Knowledge of Datafono support in the IBM ISARX25 product is assumed. Three new remote types have been introduced (they are described in detail, below):

TYPE=DFL defines a pool of pseudo-leased resources.
TYPE=DFS defines a pool of switched resources.
TYPE=DFX defines session characteristics (leased, switched, NRI table name, etc.).

This TYPE=MCH statement is used to define an HNAS logical MCH (Multi CHannel link). Inbound XOT call requests are routed to a logical MCH by the BUILD statement's RTEIN= parameter. The operands listed above are required for Datafono support, other non-Datafono specific parameters were omitted.

rmtname **REMOTE TYPE=MCH operands**	**Comments**	**CC/ VRM**
TYPE=MCH		C/240
GATE=GENERAL	Required for CTCP selection	
CTCP=(*v1,,,vn*)	Supplies Datafono CTCP values	
CUD0=(*cv1,,,cvn*)	Supplies CUD0 to CTCP mapping	
DFLNAME=(*dfl1,...,dfln*)	Identifies Leased SLU resources	
DFXNAME=(*dfx1,...,dfxm*)	Identifies Datafono SVC Options	

Note: Common remote operands not provided under this special Datafono TYPE=MCH remote type are described under the general REMOTE statement on page 4-69.

rmtname

Provides a name for the REMOTE definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF.

There are no restrictions on user assigned TYPE=MCH REMOTE names as long as the names are not defined for other remote types.

CTCP={}
CUD0={}

(changed for V2R4M0)

The **CTCP** and **CUD0** operands are used to convert a CUD0 value in an inbound call request packet to a CTCP or LLC value for an inbound call. The relationship between the CUD0 and CTCP lists is positional (GATE=GENERAL is required and the two lists must have the same length). If the CUD0 value from a call request packet is found at the i-th position in the CUD0= list then the value coded at the i-th position in the CTCP= list determines the session type as follows:

0-27 creates a GATE (LLC4) session. The value selects one of 28 CTCP Control Session LUs specified on the LUNAME= operand to receive the call request packet.

80, 83 or 85 creates an LLC0, LLC3 or LLC5 session (respectively).

100-120 creates a Datafono session and generates an index (0 through 20) used in the DFXNAME= operand to address a TYPE=DFX REMOTE statement which provides Datafono session parameters. These values are only valid if Datafono support was enabled when your distribution libraries were created.

Caution must be used if SUBADDRESS=YES is used in conjunction with LLCi= operands to establish the LLC type. Because subaddress processing precedes CUD0 processing, if a subaddress digit match is found in an LLCi= then the session type is set to LLC0, LLC3, LLC4 or LLC5 and CUD0 byte is not processed (even if it's value would have created a Datafono call). This means that if LLC5=(1) is coded then an inbound call with a subaddress digit of 1 cannot be a Datafono call.

DFXNAME=(*dfx*1,...,*dfx*n)

(new for V2R4M0)

Specifies the names of up to 21 TYPE=DFX REMOTEs used to provide Datafono session parameters. If the CTCP value is 100, the TYPE=DFX REMOTE named *dfx*1 is used. If the CTCP value is 120 the 21st TYPE=DFX REMOTE name is used. If there is no DFXNAME= entry for a given CTCP value then the call is cleared.

DFLNAME=(*dfl*1,...,*dfl*n)

(new for V2R4M0)

Specifies the names of up to 64 TYPE=DFL REMOTE statements. These statements provide a POOL of pseudo-leased resources that can be used by Datafono calls routed to this MCH. A leased resource is used when the TYPE=DFX REMOTE for the session has the XID=NO option specified (see below). If this operand is omitted, pseudo-leased resources are not supported on this MCH.

GATE=GENERAL

(changed for V2R4M0)

CUD0= and **CTCP=** require this operand (NPSI rule).

Example:

```
MCH1    REMOTE TYPE=MCH
               GATE=GENERAL
               CUD0=(22,23,FF)
               CTCP=(100,101,100)
               DFXNAME=(DFX001,DFX002)
               DFLNAME=DFL001
```

An inbound call with a CUD0 value of 22, 23, or FF is a Datafono call (associated CTCP value is in the range 100 to 120). If the CUD0 value is 22 or FF, session parameters come from the TYPE=DFX REMOTE named DFX001. If the CUD0 value is 23, session parameters come from the TYPE-DFX REMOTE named DFX002.

If either DFX001 or DFX002 specifies a call to a pseudo-leased resource (XID=NO specified) then the LU for the call will be allocated from resources created by the TYPE=DFL REMOTE named DFL001.

Datafono REMOTE TYPE=DFL Definition Statement

(new for V2R4M0)

The TYPE=DFL REMOTE definition statement is used to create a pool of pseudo-leased resources for Datafono support. The resources are available for XID=NO calls received on any MCH that addresses this DFL remote with its DFLNAME= operand.

rmtname **REMOTE TYPE=DFL operands**	**Comments**	**CC/ VRM**
TYPE=DFL		**N/240**
LUNAME=(*slunm1/plunm1,...,* *slunmn/plunmn*)	**Identifies Datafono Pseudo-Leased SLUs**	

rmtname

Provides a name for the REMOTE definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF.

There are no restrictions on user assigned TYPE=DFL REMOTE names as long as the names are not defined for other remote types. The name of this REMOTE definition statement is referenced as a ***dfl*i** value within the DFLNAME= operand of a TYPE=MCH REMOTE definition statement (see above).

LUNAME=(*slunm*1/*plunm*1,...,*slunm*n/*plunm*n)

(changed for V2R4M0)

***slunmi*/*plunm*i** specifies the HNAS SLU name and, optionally, the plu name that the SLU will communicate with.

Up to 64 names may be specified in the LUNAME= parameter. All HNAS SLU names must appear in an APPL statement in an active application major node (AMNF). The ***plunm*i** is optional. If the name is present HNAS will request a session with the PLU when HNAS activates. If the name is omitted then the PLU must acquire the HNAS SLU. Normally the plu name is provided.

Datafono REMOTE TYPE=DFS Definition Statement

(new for V2R4M0)

The TYPE=DFS REMOTE definition statement is used to create a pool of switched resources for Datafono support. The resources are available for non-XID=NO calls received on any MCH.

rmtname **REMOTE TYPE=DFS operands**	**Comments**	**CC/ VRM**
TYPE=DFS		N/240
LUNAME=(*slunm1/plunm1/idnum1,...,* *slunmn/plunmn/idnumn*)	Identifies Datafono Switched SLUs	

rmtname

Provides a name for the REMOTE definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF.

There are no restrictions on user assigned TYPE=DFS REMOTE names as long as the names are not defined for other remote types.

LUNAME=(*slunm*1/*plunm*1/*idnum*1,...,*slunmn*/*plunmn*/*idnumn*)

(changed for V2R4M0)

***slunm*i/*plunm*i/*idnum*i** specifies the HNAS SLU name, the PLU name and the IDNUM value associated with the resource. IDNUM should be coded as a 5 digit hexadecimal number.

Up to 64 names may be specified in the LUNAME= parameter on a TYPE=DFS remote. Any number of DFS remotes may be coded (there is a system wide limit of 510 REMOTEs). All DFS switched resources are placed in a common pool used for all Datafono calls.

Datafono REMOTE TYPE=DFX Definition Statement

(new for V2R4M0)

The TYPE=DFX REMOTE definition statement provides operating characteristics (leased, switched, NRI table name, etc.) for Datafono SVC calls. The DFX remote used for a particular call is determined by the CUD0 byte in an inbound call request packet used in conjunction with the CUD0=, CTCP=, and DFXNAME= parameters on a TYPE=MCH REMOTE (see above).

The OPTIONS= parameters have the same meaning as the identically named parameters on the XAI.ZGEN and XAI.ZSWTC ISARX25 statements. The ISARX25 **SWCTCP=** parameter is not presently supported.

rmtname **REMOTE TYPE=DFX operands**	**Comments**	**CC/ VRM**
TYPE=DFX		N/240
[OPTIONS=({DATAFAM|DATAF}, EMSGE, IMS, NRITAB=*name*, RETPIU, XID={NO|STD|TAB|TABSTD| (nnnnn,mmm)}]	Provides session parameters for Datafono SVCs	

rmtname

Provides a name for the REMOTE definition statement. This parameter is required and may be any valid assembler language symbol. It must also be a unique name in the CDF.

There are no restrictions on user assigned TYPE=DFX REMOTE names as long as the names are not defined for other remote types. The name of this REMOTE definition statement is referenced as a ***dfx*i** value within the DFXNAME= operand of a TYPE=MCH REMOTE definition statement (see above).

```
[OPTIONS=({DATAFAM|DATAF},
          EMSGE,
          IMS,
          NRITAB=name,
          RETPIU,
          XID={NO|STD|TAB|TABSTD|
               (nnnnn,mmm)}]
                                        (changed for V2R4M0)
```

DATAFAM|DATAF
Select one of the two values shown to specify the processing for terminal responses to type 'M' messages from the PLU.

DATAFAM
when an 'M' message is sent to a remote HNAS waits for the D(9) response from the remote. When the D(9) response arrives it is discarded and a DR+ response (if requested) is sent to the PLU. The PLU does not need logic for the D(9) message.

DATAF
when an 'M' message is sent to the remote HNAS delivers the DR+ (if requested) to the PLU as soon as the 'M' message is transmitted. When the D(9) response is received from the remote it is sent to the PLU as an input message.

The NATIVE and NATIVENV device types supported by XAI.ZSWTC define LLC0 resources. HNAS does not support these types in the OPTIONS= list. Most ISARD LLC0 support options are available in the base HNAS product. Changes may be required because functions provided by NRI tables are provided by SVC0= operands on HNAS TYPE=MCH REMOTE statements. Pseudo leased-support is not provided for HNAS LLC0 devices.

EMSGE
When this option is coded HNAS sends a 'ESPERE POR FAVOR' 'M' message if the PLU does not deliver a response in 26 seconds. If the option is not coded HNAS will not generate the message.

IMS
Code this option if the PLU is IMS.
With this parameter, instead of a DR- HNAS will send a DR+ (if solicited) and will ignore the application response in the following cases:

1) When receiving 'M' messages from the PLU and the RETPIU option is set (see below).

2) When a message of incorrect type is sent to HNAS by the PLU. A message that does not start with 'M', 'R', 'D' or 'L' has an incorrect type.

3) When a message is received outside of the normal sequence of the Datafono protocol.

If the option is not coded the above actions are not taken.

NRITAB=*name*
Specifies the name of an NRI table used to convert a calling DTE address to an IDNUM value. HNAS loads the table from a library identified by the VTAMLIB DD statement. OPTIONS=TAB|TABSTD must also be coded.

RETPIU
This option control the actions taken when a dataphone terminal session ends in an unexpected manner, i.e., without the transmission of an 'L' message. When there is an unexpected termination (e.g. clear packet received), HNAS would normally send a -RSP to the PLU and close the HNAS VTAM ACB for the session. This causes a NOTIFY PIU to be delivered to the PLU. The recovery mechanisms of the IMS will make the PLU indefinitely attempt the sending of queued responses to the terminal. When the RETPIU and IMS options are coded the following steps are taken when there is an unexpected clear:

1) The HNAS LU resource is marked ‘busy’ so it is not available for another call.

2) HNAS sends +RSP to messages from IMS and discards the messages until a non-’M’ message is received from the PLU.

3) The first message that was not sent to the remote is sent to the PLU as input. The message will be prefixed by a ‘?’ character

If the option is not coded the above actions are not taken.

XID=NO|STD|TAB|TABSTD|(*nnnnn,mmm*)
This option specifies how HNAS is to locate an SLU resource for the Datafono session. The SLU will look like the VIRTUAL=YES resources used in a NPSI environment. XID=STD is assumed if XID= is omitted.

NO
Specifies that there is no XID for the session. HNAS locates an available pseudo-leased LU from one of the TYPE=DFL REMOTEs addressed by the DFLNAME= parameter on the TYPE=MCH REMOTE with the call request. A pseudo-leased resource is available if the HNAS SLU has no session with a remote and if the LU is active (BIND, SDT received from the PLU).

STD
Locate an available LU for the session by searching the switched LU pool for a resource with an IDNUM value matching the IDNUM value in call user data bytes 1 through 3. IDNUMs are 5 (hex) digit values. The 6 digit value taken from CUD1-3 is right shifted four bits to obtain the search value. When an appropriate resource is located HNAS opens the HNAS ACB for the resource and requests a session with the PLU associated with the resource. The switched LU pool is created by TYPE=DFS REMOTE statements (see below).

TAB
Generate an IDNUM value by searching the NRI table addressed by NRITAB= for an entry with a DTE address matching the calling DTE address in the call request packet. The call is cleared if no entry is found.

TABSTD
Same as TAB except that if the IDNUM value located in the NRI table is zero then act as though STD were coded (generate IDNUM from CUD1-3).

(***nnnnn,mmm***)
Search the switched LU pool for an available LU with an IDNUM in the range nnnnn to nnnnn+mmm. nnnnn is a 5 digit hexadecimal number. mmm is a decimal number with a range of 0 to 25.

Examples:

```
DFX001  REMOTE TYPE=DFX
               OPTIONS=(DATAFAM,   ; 'M' message response processing
                        EMSGE,     ; send please wait messages
                        IMS,       ; PLU IS IMS
                        RETPIU,    ; return 1st PIU not sent
                        XID=NO)    ; allocate leased resource from
                                   ; TYPE=DFX REMOTE addressed
                                   ; DFLNAME= parameter


DFX002  REMOTE TYPE=DFX
               OPTIONS=(XID=STD)   ; allocate switched resource from
                                  ; switched resource pool created by
                                   ; TYPE=DFS REMOTE statements.
                                   ; use 5 digits from CUD1 to CUD3
                                  ; to supply the IDNUM to locate the
                                   ; resource.
```

END Definition Statement

The END definition statement is used to delimit the end of the X.25 Host NAS configuration. When the END definition statement is encountered, all remaining HNAS defaults are set and all required storage areas are allocated. The END definition statement is optional. If omitted, the CDF end of data (EOD) exit (via the DCB EODAD parameter) will treat the EOD condition as though the END definition statement was present.

[*symbol***]**	**END**	**operands**

[*symbol***]**

Provides a name for the END definition statement. This parameter is optional and may be any valid assembler language symbol.

CHAPTER 5

Migration

Chapter 5 - Migration

This chapter describes the configuration operands and run time functions that have changed or been removed under the denoted HNAS releases. The specific migration list items should be carefully reviewed before implementing a product upgrade or refresh to ensure that your environment isn't impacted by the changes.

Migration Overview

Every effort has been made to preserve previously defined default values so that migration to newer levels of HNAS occurs with minimal impact to our customers. There are times, however, when the program logic must be updated or altered for new program options or to correct previously defined default assumptions found to be incorrect. This includes changes to configuration operands, messages and runtime processes that can cause the environment to operate differently than previous releases. The migration sections also contain information on documentation section reassignment, as appropriate.

As each new release is introduced the migration issues for that particular release will be denoted under the respective V*n*R*n*M*n* release heading **"Program Logic changes to V*n*R*n*M*n* that may affect your environment migration"** of this documentation area.

When upgrading to a newer release, you should review all V*n*R*n*M*n* migration sections after your existing release up through the release you are interested in upgrading to.

> If you are upgrading from V2R3M0 to V2R4M0, you need only review the V2R4M0 migration section.
>
> If you are upgrading from V2R2M0 to V2R4M0, you need to review the V2R2M0 thru V2R4M0 migration sections.
>
> We also recommend that you review the migration section for the release that you are upgrading from to see if any items were added to the migration list after your HNAS product implementation date. This doesn't occur often because APARs are typically issued to document these type of changes for an active release.

We suggest that you review the **DNAS** console command output to determine the current release level and generation date of your existing HNAS product. The information is normally available in the SYSPRINT area when HNAS is started although it can be displayed at any time using the Local or Remote DNAS console command.

For **V2R4M0**, the DNAS command produces the following output:

```
HNAS --> VERSION=V2R4M0 DIST=SMP                                    1
         ASMDATE=2006/01/02 ASMHOST=ZOS                             2
         RUNNING UNDER z/OS 01.04.00                                3
         DNAS COMMAND ENTERED AT 01:21:33 ON 2006/01/03             4
         HNAS PROGRAM STARTED AT 01:09:54 ON 2006/01/03             5
         HNAS PRODUCT CREATED AT 21:17:43 ON 2006/01/02             6
         CREATED WITH MAINTENANCE THROUGH APAR 2400000              7
         MOST RECENT MAINTENANCE APPLIED IS APAR 2400000            8
         SHIPID=0000000000199999 AUTH=00                            9
         CUSTMAC=                         <-- if present           10
         CUSTOBJ=                         <-- if present           11
         CUSTID= cid_cust#  (i.e. trl_99999)                       12
         CUSTINFO= optional customer info                          13
         MAINTENANCE/USE ANNIVERSARY DATE IS 2007/01/31            14
         DATAFONO SUPPORT IS INCLUDED     <-- if present           15
                                                                   16
         APARID   MODULE  (APPLIED MAINTENANCE)                    17
         ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400000       18

        - or -

         aparid  NOT INSTALLED                                     19
         :
         aparid  NOT INSTALLED                                     n
```

Items in **bold** are new for 240.

Line numbers to the right of the DNAS display above are for reference only. Lines 10 and/or 11 are displayed only if a custom MACLIB and/or OBJLIB were used to create the HNAS distribution. The APAR number in line 7 represents the highest APAR number that was included in the HNAS distribution when it was created (*distribution APAR*). The APAR number in line 8 represents the highest APAR number applied to HNAS after the distribution was installed at the customer site (*applied APAR*). The *distribution APAR* and the *applied APAR* will be the same if no maintenance has been applied since the original HNAS install. If there are no missing APARs between the *distribution APAR* and the *applied APAR* (all APAR numbers are consecutive), line 18 will be displayed. If, however, there are 'wholes' between the *distribution APAR* and the *applied APAR* then the missing APAR(s) are displayed following line 17 of the form shown in line 19.

Items in **bold** are new for 240. Please refer to the DNAS section of the Console Sub-system Operations Guide for a description of the new fields.

For **V2R3M0**, the DNAS command produces the following output:

```
HNAS --> VERSION=V2R3M0 DIST=NON-SMP ASMDATE=12/16/05 ASMHOST=ZOS  1
         RUNNING UNDER z/OS 01.04.00                               2
         STARTED AT 10:54:50 ON 12/16/2005                         3
         CREATED AT 10:11:31 ON 12/16/2005                         4
         CREATED WITH MAINTENANCE THROUGH APAR 2300168             5
         MOST RECENT MAINTENANCE APPLIED IS APAR 2300168           6
         SHIPID=9999999999999999 AUTH=00                           7
         CUSTMAC=COMM1.TEST.HNASMAC                                8
         CUSTOBJ=COMM1.TEST.HNASOBJ                                9
                                                                  10
         APARID   MODULE  (MAINTENANCE STATUS)                    11
         ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2300168      12

        - or -

         aparid  NOT INSTALLED                                    13
         :
         aparid  NOT INSTALLED                                    n
```

Line numbers to the right of the DNAS display above are for reference only. Lines 8 and/or 9 are displayed only if a custom MACLIB and/or OBJLIB were used to create the HNAS distribution. The APAR number in line 5 represents the highest APAR number that was included in the HNAS distribution when it was created (*distribution APAR*). The APAR number in line 6 represents the highest APAR number applied to HNAS after the distribution was installed at the customer site (*applied APAR*). The *distribution APAR* and the *applied APAR* will be the same if no maintenance has been applied since the original HNAS install. If there are no missing APARs between the *distribution APAR* and the *applied APAR* (all APAR numbers are consecutive), line 12 will be displayed. If, however, there are 'wholes' between the *distribution APAR* and the *applied APAR* then the missing APAR(s) are displayed following line 11 of the form shown in line 13.

Note: The **DNAS** conditional output for display line 11 heading '**APARID MODULE (MAINTENANCE STATUS)**' and display line 13 (and what follows) '*aparid* **NOT INSTALLED**' were originally introduced into 230 via APAR 2300062. The text, as it now appears, was subsequently changed by 2300089.

Note: The **DIST=NON-SMP|SMPE**, **CUSTMAC=** and **CUSTOBJ=** fields were introduced into 230 via APAR 2300112.

We also suggest that anytime problems arise, you issue the **DNAS** console command and forward the output along with your **CDF** (configuration data file) to your HNAS Sales and Support representative when ordering the product refresh or upgrade. This will provide the HNAS support representative with an opportunity to review your specific configuration and maintenance environment in an effort to provide a smooth transition to the newer product level.

Distribution Library Names

We strongly recommend that all HNAS product refreshes and upgrades be installed into new uniquely defined libraries. We suggest that the second level dataset name qualifier identify either the:

@*vrmnnnn* (@**2400001**) HNAS Version, Release, Modification *vrm* and APAR maintenance level id *nnnn*, or

V*n*R*n*M*n* (**V2R4M0**) HNAS release designation and a single digit Generation Identifier (**A**, B,,, to clearly identify the iteration level)

in an effort to clearly identify the maintenance level for product upgrades or refreshes.

Samples: *hlq*.@**2400001**.HNASMAC
hlq.**V2R4M0A**.HNASMAC

Refer to Chapter 6 section 'Upgrade and Refresh Distribution Maintenance Installation' section of this documentation manual for additional information regarding maintenance installation topics.

Information Sources

For additional information, please refer to the following information sources:

1) "**HNAS New Features - V*n*R*n*M*n***" located in the Appendix D section of this documentation manual;

2) Web sites: **www.comm-pro.com/hostnas/maint/index.htm**
www.comm-pro.com/hostnas/docs/docindx.htm

3) **README**, **@README** or **@CSTMEMO** files located in with the HNAS product distribution (these optional members are located in the **HNASMAC** or **HNASMACX** macro libraries) or

4) "**Document, V*n*R*n*M*n***" and/or "**Program, V*n*R*n*M*n***" entries are located in the **Revision Index** section of the Special Revision Master Index book (available upon request)

Please refer to Chapter 6 of this documentation for information on maintenance.

Migration - V2R4M0

Program Logic changes to V2R4M0 that may affect your environment migration

1) In an effort to standardize the date format in HNAS documentation, console command output and alert messages date fields, the HNAS generated date format are now in the yyyy-mm-dd date format while the Host system generated dates are provided in their native format which is typically in the month day, year (January 1, 2006) format.

   In addition, throughout our documentation the dataset name qualifier of **QQQQ.** or **qqqq.** has been replaced with **hlq.** to better reflect the **h**igh **l**evel **q**ualifier of the dataset names.

2) The HNAS non-SMP/E product installation process has changed. Several improvements were implemented including a REXX exec that builds the installation JOB from a template parameter file and reduces the steps required to install the product.

   The sample **ZAP** step was removed from the **HNASINST** JOB and is no longer provided or necessary in the **HNASMNT** JOB (the ZAP step was historically provided as a sample although all HNAS maintenance is provided in macro or object format, no ZAPping is required). In addition, the HNAS product installer no longer has to manually create the non-SMP/E **HNASMNT** JOB because the 240 installation process now does this automatically.

3) The **DNAS** command output display contains additional environment information, some of the display output lines were reordered. Please refer to the DNAS command description in the Console Guide for additional information.

4) The **PRNTXTP ON** parameter option now defaults to **PRNTXTP OFF**. Other **PRNTtype** start parameters were under consideration to be changed from **ON** to **OFF** defaults although further review revealed that important Information messages would be eliminated from SYSPRINT. The best way to eliminated unwanted **NAS*cnnn*I** SYSPRINT messages is to enable the ALRMFLTR=(...,**NAS*cnnn*I(P)** Purge filtering.

   The **PRNTtype** start parameter controls the logging of **I**nformational **NAS*cnnn*I** (severity ***s-value*** of **I** are controlled by this process) SYSPRINT alarm and alert message activity.

5) Message content was altered (improved) in the following Alert Messages:

   NAS0030I, **NAS0050A**, **NAS0200W**, **NAS0201E (**was **NAS0201W), NAS0202W NAS7701W**, **NAS7702E**, **NAS7717W** and **NAS9205*i***.

   Please refer to the respective message ID for additional information.

6) The following messages are now written to SYSPRINT regardless of the PRNTtype or SHOWtype setting via DEV=F (see Informational Alert Message Considerations).

   NAS0030I and **NAS9205I**.

Please refer to the Messages & Codes Alert Message sections for additional information concerning these messages and other DEV=F messages.

7) MCH names **SKIP** and **CLEAR** are now reserved names for **RTEIN=** list filter processing to allow predefining of dteaddr entries to be positionally skipped (reserved for future use) as well associating dteaddr values filtering for clear processing. Please refer to the remote type LOCAL, RTEIN= parameter for additional information.

8) The **TRCMCH ALLON** console command no longer sets the ICR, ICRF, OCR, ICLR and OCLR options in order to force it to behave the same as the TRCMCH ALLON start parameter. The ALLON function is now separate from the ICR, ICRF, OCR, ICLR and OCLR trace requests. The TRCMCH ALLON start parameter and console command will now start MCH tracing only although the TRCMCH ICRF option is now set by default. When the ICR, ICRF, OCR, ICLR and/or OCLR trace options are in effect, corresponding trace entries are only logged if the MCH(s) are also being traced (TRCMCH ALLON is in effect).

   The console command or start parameter operators **ICR, ICRF, OCR, ICLR** and **OCLR** can still be entered individually or as a group. While the start parameter decode assumes that these operators are *global*, the console command processor now requires the **GBL** option (or the assumed GBL default mentioned below) to identify these parameters as *globa*l. If **GBL** is omitted and RNM=*rmtname* **or** ID=*nn* is specified, these parameters are assumed to be *local* to the MCHs identified by the RNM= **or** ID= modifiers. If GBL is omitted and both RNM= **and** ID= are not set (null), these parameters are assumed to be *global* (**GBL** is assumed).

   Because of the complexity of the TRCMCH command, we recommend that when working with command operators (**ICR, ICRF, OCR, ICLR** and **OCLR**) in global mode that the **GBL** parameter always follow the primary command (i.e. **TRCMCH GBL ICR ON|OFF**).

9) Console commands (whether entered via CONCMDQ= or by the console operator) will now be prefixed with text "**HNASCMD->**" when written to the SYSPRINT output in an effort to improve general readability and improve find|search capabilities.

10) Some **TRC*type*** console commands (TRCLU, TRCMCH, TRCMCHX and TRCVC) will now display "**HNASCMD->**" identifying the command and related command **modifiers** and **arguments** (including default arguments) that will be used during command execution. Please refer the 240 New Features or the specific console command for additional information.

11) The **DLU** and **DVC** command output display now contains heading field SESSINIT which identifies the VC connect state as well as the initiator (direction) of the Call Request. The CID field is no longer provided in the default (FMT1) base display although it is provided in the 'DLU|DVC FMT2' or 'DLU|DVC SHOWCID' display. Please refer to the 240 New Features section for additional DLU or DVC improvements.

12) The PRTLMT= operand maximum value was changed from 134217727 (X'7FFFFFF) to 16777215 (X'FFFFFF') for PRTSWLST=DYNAMIC support via 240 as Enhancement APAR 2400001. This was required because the DYNALLOC parameter list element for the SYSOUT print limit (SVC99 key X'001B') is a 3-byte field.

13) The HNAS buffer size has been increased by 16 bytes for diagnostic purposes. This means that when you specify a BFRSIZ= value, you must allow for this new 16 byte area.

Prior to V2R4, the packet header length was 52 bytes. Thus, for a packet size of 128, BFRSIZ=180 had to be specified if a single buffer was to contain a full packet.

Starting with V2R4, the packet header is now 68 bytes. Thus, for a packet size of 128, BFRSIZ=196 has to be specified if a single buffer was to contain a full packet.

Beginning with **APAR 2400002**, HNAS will automatically increase the BFRSIZ= value by 16-bytes and issue the following message if the BFRSIZ= value does not already compensate for the new packet header length:

```
NAS1121D BUILD BFRSIZ=dddd AUTOMATICALLY INCREASED TO dddd,
         DEBUG FIELD ADDED
```

The BFRSIZ= value will not be increased by 16-bytes and this message will be withheld if the BFRSIZ= value already accounts for the new 68 byte packet header, that is, it is one of the following values:

196 = 68 + 128
324 = 68 + 256
580 = 68 + 512
1092 = 68 + 1024
2116 = 68 + 2048
4164 = 68 + 4096
9362 = TCPIP transfer limit

Note that the packet size is always given as a power of 2 via facility code 42 therefore intervening values do not have to be tested.

Note: The logic to automatically increase the BFRSIZ= was introduced into 240 under APAR 2400002.

14) The following console error messages have been changed:

from: `NASC522E INPUT DATA INVALID, REQUIRED`
to: `NASC532E PARAMETER DATA INVALID, REQUIRED`

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

Note: These error message changes were introduced into 240 under APAR 2400014.

15) APAR **2400018** corrected **IDLETO=** processing so that the session idle timeout value is taken from the IDLETO= parameter on the MXT (if coded), the SPU (if coded), the MCH (if coded) and BUILD (if coded - 0 is the default for BUILD). Prior to the APAR if, for

example, an SVC0= string addressed a TYPE=MXT REMOTE with an omitted IDLETO= parameter then a value of zero was used instead of the MCH or BUILD IDELTO= values.

16) Prior to APAR 2400017, only the following alarm messages were forced and some, but not all, could be filtered from SYSCONS and SYSPRINT using the BUILD ALRMFLTR= operand.

Message ID	Message Description	Subject to ALRMFLTR
NAS0001I	HNAS initialization complete alert message	NO
NAS003*x*I	HNAS shutdown messages	NO
NAS0205I	SYSPRINT logging terminated by SHUTDOWN	NO
NAS0208I	PRTSWLST= SYSPRINT log dataset opened	NO
NAS0210I	PRTSWLST= SYSPRINT log switch requested	NO
NAS0910I	Date crossover message issued at midnight	NO
NAS3796I	Gate Fast Connect LU session start alert message	YES
NAS3798I	LU session start alert message	YES
NAS3799I	LU session end alert message	YES
NAS920*x*I	HNAS authorization messages	NO

Effective with APARs 2400017 and 2400019, the following alarm messages are now forced and some, but not all, can be filtered from SYSCONS and SYSPRINT using the BUILD ALRMFLTR= operand. Note that existing forced message IDs prior to APAR 2400017 are listed in **bold**.

Message ID	Message Description	Subject to ALRMFLTR
NAS0001I	**HNAS initialization complete alert message**	**NO**
NAS003*x*I	**HNAS shutdown messages**	**NO**
NAS0050A	Tracing suspended or snapshot dump taken by *consname*	NO
NAS0060W	Tracing resumed or snapshot dump resumed by *consname*	NO
NAS0070W	Trace trap action status changed by *consname*	NO
NAS0071W	Trace trap action status	NO

Message ID	Message Description	Subject to ALRMFLTR
`NAS0120I`	SYSCONS error and info alarms enabled by *consname*	NO
`NAS0121W`	SYSCONS non-forced alarms disabled by *consname*	NO
`NAS0122I`	SYSCONS error alarms enabled by *consname*	NO
`NAS0130W`	SYSCONS alarm options modified by *consname*	NO
`NAS0201E`	SYSPRINT log record limit reached, logging terminated	NO
`NAS0204E`	SYSPRINT ABEND, logging terminated	NO
`NAS0205I`	**SYSPRINT logging terminated by SHUTDOWN**	**NO**
`NAS0207W`	SYSPRINT PRTSWLST= log datasets all used	NO
`NAS0208I`	**SYSPRINT PRTSWLST= log dataset opened**	**NO**
`NAS0209E`	SYSPRINT PRTSWLST= log dataset could not be opened	NO
`NAS0210I`	**SYSPRINT PRTSWLST= log switch requested**	**NO**
`NAS0210I`	SYSPRINT logging enabled by *consname*	NO
`NAS0210W`	SYSPRINT trace logging enabled by *consname*	NO
`NAS0211I`	SYSPRINT trace logging disabled by *consname*	NO
`NAS0211W`	SYSPRINT logging disabled by *consname*	NO
`NAS0230W`	SYSPRINT logging options modified by *consname*	NO
`NAS0910I`	**End of day crossover message issued at midnight**	**NO**
`NAS3796I`	**Gate Fast Connect LU session start alert message**	**YES**
`NAS3797I`	SLU BIND received from PLU	YES
`NAS3798I`	**LU session start alert message**	**YES**
`NAS3799I`	**LU session end alert message**	**YES**
`NAS5720I`	Datafono LU session start alert message	YES
`NAS5721I`	Datafono leased LU normal session end alert message	YES
`NAS5722W`	Datafono leased LU error session end alert message	YES
`NAS920xI`	**HNAS authorization messages**	**NO**

17) Effective with APAR 2400028, the following configuration error messages have been substituted when the *vclmt* value for the PVC=, SVC0=, SVC3=, SVC4= or SVC5= operands exceed their respective operand array limit:

```
from: NAS1321E ERROR: REMOTE   SVC0=(512,
  to: NAS1303E LIMIT: REMOTE   SVC0=(512,
```

Note that the maximum *vclmt* values are as follows:

```
PVC   SVC0   SVC3   SVC4   SVC5
255    511    255    511    511
```

The NAS1303E message will also be issued when the *vclmt* value for the alternate syntax for SVC0=, SVC4= or SVC5= exceeds the respective operand array limit. For example, if SVC0=(MCH1XX,01,256) is specified, the following message will be issued when the *vclmt* suboperand is encountered:

```
NAS1303E LIMIT: REMOTE   256)
```

This occurs because the generated SLU names are limited to eight (8) characters in length (MCH1XX01 through MCH1XXFF). The *vclmt* value of 256 would cause SLU names to exceed the 8 character limit (MCH1XX100).

Note: These error message changes were introduced into 240 under APAR 2400028.

18) Effective with APAR 2400031, the NAS3711 alert message, which reports that HNAS has received a -RSP from the PLU, can be issued with ‘I’ and ‘W’ severity levels. The ‘W’ severity is used when HNAS will terminate the VTAM session based on the sense data. The ‘I’ severity indicates that the VTAM session will be left active. In addition, the sense data sent by the PLU is now displayed in the alert.

19) Effective with APAR 2400032, the following error messages will be issued if an invalid table is named for the LOGTAB=, USSTAB= or OPTIONS=NRITAB= operands:

<u>For BUILD</u>:

```
NAS1111E BUILD LOGTAB=lgtbnm IS NOT A VALID LOGON TABLE, REQUIRED
NAS1111E BUILD USSTAB=ustbnm IS NOT A VALID USS TABLE, REQUIRED
```

<u>For REMOTE</u>:

```
NAS1311E REMOTE rmtnm LOGTAB=lgtbnm IS NOT A VALID LOGON TABLE, REQUIRED
NAS1311E REMOTE rmtnm USSTAB=ustbnm IS NOT A VALID USS TABLE, REQUIRED
NAS1311E REMOTE rmtnm NRITAB=nrtbnm IS NOT A VALID NRI TABLE, REQUIRED
```

Because these messages are severity ‘E’, HNAS will terminate after the CDF is completely scanned if any of these messages are issued.

An invalid table is one that does not conform the format as dictated by the given operand. For example, if the LOGTAB= operand specifies a USS table name instead of a LOGON table name (e.g., LOGTAB=ISTINCDT), the following message will be issued:

```
NAS1311E REMOTE rmtnm LOGTAB=ISTINCDT IS NOT A VALID LOGON TABLE, REQUIRED
```

Prior to this APAR, an invalid table was not detected until the table was selected during call setup processing.

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`

The new configuration messages now mean that a bad table will be detected during CDF scan processing so they can be corrected before HNAS actually is allowed to start.

20) Effective with APAR 2400032, most console error message will now include the name of the command that generated the message. For example:

`NASC013E UNSUPPORTED FUNCTION,` *`cmdname`* `COMMAND ABORTED`

21) Effective with APAR 2400034, the EXEC command will stop queuing commands when an embedded EXEC command or the new END statement is decoded. This will reduce the likelihood of receiving the following error message:

`NASC054E EXEC CMDLIST1 FILE EXCEEDS QUEUED COMMAND LIMIT, EXEC COMMAND ABORTED`

If more than 512 bytes of command data and data length bytes are listed in the command list identified by the *ddname* argument, the NASC054E message will be generated. Prior to APAR 2400034, this would occur even if the command list contains an embedded EXEC command (which ends the current command list) listed early in the command list file. The problem occurred because the decode of the embedded EXEC command was not performed until after the entire command list was read into memory. Some users like the ability to list many commands in an EXEC command list beyond the EXEC command for use at a later time. The old logic prevented large numbers of saved commands to be listed in a command list file. The new logic will allow any number of commands to be saved after the END or embedded EXEC command.

Note: EXEC command list END statement and suspend queuing support was introduced into 240 as Enhancement APAR 2400034.

22) Effective with APAR 2400039, PING response alarm messages no longer contain the name of the console that initiated the request. The messages now contain the destination IPADDR and DTEADDR addresses in all PING response alert messages as well as providing a correlation sequence number in all PING request and response messages.

23) Effective with APAR 2400053, the timer value used for the PVC VTAM connect timer has been made a constant value of 60 seconds. This timer controls how often HNAS will issue a REQSESS VTAM macro asking the PLU for a BIND to start a session for a PVC that has completed the XOT PVC SETUP process. Previously the value used depended on circumstances -- for example, if a request failed because the PLU was not active (SENSE=08570001) the value was 2 minutes. This means that the NAS3702W alert (REQSESS FAILED) may be observed more often and that the sessions will be started in a more timely fashion when the PLU is activated. This APAR also provides improved

DIAGX= values for the 196 (C4) Reset Diagnostic code in NAS5705W RESET SCHEDULED messages.

24) Effective with APAR 2400055, the following HNAS messages were altered:

NAS2511M monitor message now contains a retry counter, changes in **bold**:

```
NAS2511M XOT TAP TIMEOUT, RESPONSE NOT RECEIVED FOR CONNECTION
         SETUP (nnnnn)
```

NAS2502E alert message now contains processing options, changes in **bold**:

```
NAS2502E ROUTER CONTACT LOST, CLOSEONTAPFAILURE OPTION IS IN
         EFFECT
```

25) Effective with APAR 2400074, the following HNAS messages were altered:

NAS7708W alert message now contains the LU name, changes in bold:

```
NAS7708W XOT PVC SETUP INIT=ininm     PVC=pvc#
         RESP=rspnm    PVC=pvc# LU=luname
```

NAS7718T alert message now contains the LU name, changes in bold:

```
NAS7718T ii.iii.iii.iii(port) {CALL REQ | PVCSETUP}
         TO MCH mchname LU luname
```

26) Effective with APAR 2400075, HNAS will now only accept queued command strings up to 63-bytes in length instead of 80-bytes in length. This change was required to allow two bits in command string count byte to be reserved for flags.

This change affects the CONCMDQ= and SCHEDULE= configuration parameters as well as the EXEC and SCHEDULE console commands.

27) Effective with APAR 2400077, the configuration will now issue the NAS1311S message when SOCLMT=*limit* and the sum of the TYPE=XOT REMOTE VCLMT= values (for which the LOCAL is HOME) is greater than *limit*-2. This eliminates a runtime ABEND during HNAS initialization. In the past, the test was for *limit* only which did not take into consideration 2 reserved sockets. One for the LOCAL's HOME IP address (the LISTENing socket) and one for a temporary socket that is used during REMOTE connection processing.

The NAS1311S message will now have the following format:

```
NAS1311S REMOTE XOTCNOT2 HOME XOTSRVR LIMIT REACHED
                SOCCNT=00303 SOCLMT=00602 NEEDED=00301
```

This message is issued when NEEDED+SOCCNT>SOCLMT. SOCCNT includes the 2 reserved sockets.

28) Effective with APAR 2400078, the HNAS DLU STAT column heading was renamed to LUST. VCTO column was added to the DVC output and LUTO column was added to the DLU output. See command descriptions for specifics.

29) Effective with APAR 2400079, the following configuration and alert message changes have been made:

1) The severity code for the following configuration messages have been changed from W to I to prevent CC=4 from being set.

```
NAS1321W REMOTE IPADDR WAS ALSO SPECIFIED FOR LOCAL
         NAMED LXOT
NAS1321W VALID ONLY IN HNAS-TO-HNAS ENVIRONMENT USING THE
         SAME TCPIP STACK

NAS1321W REMOTE IPADDR AND PORT WERE ALSO SPECIFIED FOR
         REMOTE NAMED R2CNOT1
NAS1321W THE SOCKETS FROM BOTH REMOTES WILL BE COMBINED
         IN A COMMON POOL
```

Message ID **NAS1321I** will now be used for these configuration messages. This change was made because these messages are actually more for information than as a warning.

2) The following configuration messages are now issued only during a FASTRUN execution.

```
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH1P001 (DEFINED ON
         MCH MCH1)
NAS1391I AS PVC ENTRY 000) COME FROM MXT MXTPVC01 RATHER THAN
         THE MCH
NAS1391I --> FAC=420808430404
```

This change was made to reduce the number of messages logged in SYSPRINT during a normal HNAS run. It also minimizes the possibility of receiving the NAS0200W
message.

3) The severity code for the following alert message has been changed from W to I to prevent CC=4 from being set.

```
NAS0200W 00016384 SYSPRINT LOG RECORDS WRITTEN
```

Message ID **NAS0200I** will now be used for these configuration messages. This change was made because these messages are actually more for information than as a warning.

30) Effective with APAR 2400080, the CONCMDQ= operand will now accept either an inline command list or a DDNAME that identifies a command list. In order to differentiate

between an inline command list and a DDNAME, the command list must be specified as a sublist (enclose in parentheses). A DDNAME is assumed if the CONCMDQ= value is not in parentheses. Users must be aware of this when they have previously coded a single command as a CONCMDQ= value. For example, CONCMDQ=DNAS, which in the past would have queued the DNAS command for execution, will now treat DNAS as a DDNAME. The correct syntax is as follows:

CONCMDQ=*ddname*|(*cmd*1,...,*cmd*n)

Note: The EXEC= operand was added as an alias for the CONCMDQ= operand to provide symmetry with the SCHEDULE= operand and SCHEDULE console commands. The EXEC console command which was previously the counterpart for the CONCMDQ= operand is now also the counterpart for the EXEC= operand.

31) Effective with APAR 2400083, corrections have been made in the way HNAS handles the omitted SVRSTRT start parameter when the TCPIP stack is deactivated.

Problem 1:
When the SVRSTRT option is not in effect (the default when not specified), **prior to this APAR**, HNAS would remain active after the stack was taken down when it should have been shutdown.

The following callout message could be generated because HNAS erroneously would not reconnect with the stack even though VTAM sessions would still be allowed to come active.

```
NAS7720W BRN1 CALL OUT, CAN'T CALL CALLED
         ADDR=F311022081263127 CALLING ADDR=C311022081040000
```

The following TCPIP alerts would also be present:

```
NAS2401W CLIENT=169.165.250.144(01656) SOCKID=0001
         PCEID=0009 NAME=BR1XOT
NAS2401W RECEIVE REQUEST FAILED, RC=FFFFFFFF 00000005 (00005)
```

This implied that a TCPIP RECEIVE operation ended with I/O errors.

```
NAS2401W CLIENT=169.165.250.144(01657) SOCKID=0002
         PCEID=000A NAME=BR1XOT
NAS2401W RECEIVE REQUEST FAILED, RC=FFFFFFFF 0000040C (01036)
```

This implied that a TCPIP RECEIVE operation ended because the stack was down.

```
NAS2102E TCPIP SEVER INDICATED, PATHID=0001 TCPNAME=RCCTCP11
```

This says TCPIP stack was taken down.

Note: Inbound calls would also fail (calls cleared) under this scenario because the HNAS HOME server component remained down.

Effective with APAR 2400083, when the SVRSTRT option is not specified and the stack is taken down, HNAS will shutdown as the documentation specifies. This will require HNAS to be restarted when the stack is restarted.

Note: If the SVRSTRT option had been specified, HNAS would have monitored the stack for reactivation which would have prevented the callouts and callins from failing. This still applies even with APAR 2400083 on the system.

Problem 2:
A second problem has also been fixed when the SVRSTRT option is omitted and multiple stacks are defined to HNAS. In this case, connections to an active stack would hang if one stack was stopped.

When SVRSTRT was omitted and only one stack is defined to HNAS, HNAS should shutdown (see problem 1). Due to a logic error, HNAS remained active but would not allow the stopped stack to restart. In addition, the omitted SVRSTRT logic erroneously prevented other stacks from continuing operation. This caused the hang condition.

HNAS has been modified to force the SVRSTRT option to be in effect when multiple stacks are specified. This will ensure that an active stack will continue to communicate with HNAS if another stack is taken down. In addition, it will allow the deactivated stack to come active again. The following new configuration message is generated in this case.

```
NAS1211D MULTIPLE TCPNAME VALUES SPECIFIED, SVRSTRT OPTION FORCED
```

Note: These 2 problems were introduced into HNAS when multiple server support was added in February of 2002. During our testing of the multiple server support, the SVRSTRT option was in effect and so HNAS recovered as it was supposed to when one stack was taken down. We believe that these problems were not discovered until recently because most customers run with the SVRSTRT option active or manually shutdown HNAS whenever a stack is taken down. In either case, it prevented these problems from being observed until now.

In addition to these problem fixes, APAR 2400083 includes format changes for a few HNAS alert messages which will now be forced to SYSCONS and cannot be filtered. These alert messages are as follows:

Old: `NAS2102E TCPIP SEVER INDICATED, PATHID=0001 TCPNAME=TCPIPROD`

New:
```
NAS2102E SERVER=010.117.056.171(01998) SOCKID=0000
         PCEID=0007 NAME=LXOT
NAS2102E SEVER INDICATED FOR TCPNAME=TCPIPROD PATHID=0001
```

Old: `NAS2103W TCPIP QUIESCE INDICATED`

New:
```
NAS2103W SERVER=010.117.056.171(01998) SOCKID=0000
         PCEID=0007 NAME=LXOT
NAS2103W QUIESCE INDICATED FOR TCPNAME=TCPIPROD
```

Old: `NAS2104I TCPIP RESUME INDICATED`

New: NAS2104I SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2104I RESUME INDICATED FOR TCPNAME=TCPIPROD

Old: NAS2105S CLIENT=010.117.056.100(20944) SOCKID=001E
PCEID=0009 NAME=R1CNIN
NAS2105S TCPIPROD TRANSFER FAILED, RC=FFFFFFFF 0000040C (01036)

New: NAS2105S CLIENT=010.117.056.100(20944) SOCKID=001E
PCEID=0009 NAME=R1CNIN
NAS2105S TRANSFER TO TCPNAME=TCPIPROD HAS FAILED,
RC=FFFFFFFF 0000040C (01036)

Old: NAS2030I SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2030I API CONNECTION TO TCPIPROD VR=0614 CAN BE PERFORMED

New: NAS2030I SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2030I API CONNECTION TO TCPNAME=TCPIPROD VR=0614
CAN BE PERFORMED

Old: NAS2031W SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2031W API CONNECTION TO TCPIPROD IS BEING DEFERRED

New: NAS2031W SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2031W API CONNECTION TO TCPNAME=TCPIPROD IS BEING DEFERRED

Old: NAS2032E SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2032E API CONNECTION TO TCPIPROD CANNOT BE PERFORMED,
RC=FFFFFFFF 0000040C (01036)

New: NAS2032E SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2032E API CONNECTION TO TCPNAME=TCPIPROD CANNOT BE
PERFORMED, RC=FFFFFFFF 0000040C (01036)

Old: NAS2050I SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2050I API CONNECTION TO TCPIPROD HAS BEEN ESTABLISHED

New: NAS2050I SERVER=010.117.056.171(01998) SOCKID=0000
PCEID=0007 NAME=LXOT
NAS2050I API CONNECTION TO TCPNAME=TCPIPROD HAS BEEN
ESTABLISHED

```
Old: NAS2051S SERVER=010.117.056.171(01998) SOCKID=0000
              PCEID=0007 NAME=LXOT
     NAS2051S API CONNECTION TO TCPIPROD HAS FAILED,
             RC=FFFFFFFF 0000040C (01036)

New: NAS2051S SERVER=010.117.056.171(01998) SOCKID=0000
              PCEID=0007 NAME=LXOT
     NAS2051S API CONNECTION TO TCPNAME=TCPIPROD HAS FAILED,
              RC=FFFFFFFF 0000040C (01036)

Old: NAS2060I SERVER=010.117.056.171(01998) SOCKID=0000
              PCEID=0007 NAME=LXOT
     NAS2060I API CONNECTION TO TCPIPROD HAS BEEN TERMINATED

New: NAS2060I SERVER=010.117.056.171(01998) SOCKID=0000
              PCEID=0007 NAME=LXOT
     NAS2060I API CONNECTION TO TCPNAME=TCPIPROD HAS BEEN
              TERMINATED

Old: NAS2061S SERVER=010.117.056.171(01998) SOCKID=0000
              PCEID=0007 NAME=LXOT
     NAS2061S API DISCONNECTION FROM TCPIPROD HAS FAILED,
             RC=FFFFFFFF 0000040C (01036)

New: NAS2061S SERVER=010.117.056.171(01998) SOCKID=0000
              PCEID=0007 NAME=LXOT
     NAS2061S API DISCONNECTION FROM TCPNAME=TCPIPROD HAS FAILED,
              RC=FFFFFFFF 0000040C (01036)
```

32) Effective with APAR 2400084 GENERIC resource names may be specified when PCNE (LLC0) or PAD (LLC5) callout resources are defined (SVC0=, SVC5=). This allows a PLU to acquire the HNAS resource using a generic name.

33) Effective with APAR 2400085 for trail distributions only, when the DNAS command output shows a 16-digit SHIPID= field that starts with a 1 (1xxxxxxxxxxxxxxx) and a non-zero AUTH=value, it indicates that HNAS will ABEND when the trial NASAUTH file expires. Conversely, if the SHIPID= field is not 16 digits in length, HNAS will continue to run when the trial NASAUTH file expires but will not be allowed to restart if it is shutdown.

34) Effective with APAR 2400086 utility changes were made to the HNAS Authorization Date validation process to correct an error observed in the trial product expiration date processing.

35) Effective with APAR 2400089, various utility changes were made to HNAS.

The following is a summary of these changes:

1) NAS9206I ... ALLOWS DATAFONO SUPPORT message reassigned to NAS9220I

2) HNAS will now generate End of Maintenance/Use Anniversary Date countdown messages NAS9206i

3) Alarm Log Table Expansion and Relocation so that all messages are logged when HNAS initialization begins.

Because the alarm logging table is allocated after the CDF scan completes, a number of runtime alarms (like NAS92*xxs* AUTH alarms) were not logged. HNAS has been modified to allocate the alarm logging table immediately after it starts with a default entry limit of 512. The limit, which was previously specified in the CDF as the ninth (last) suboperand of the ALRMLMTS= operand, has been retired. If a value is specified, the following configuration alert message is generated:

```
NAS1101D BUILD ALRMLMTS LOGGING TABLE LIMIT PARAMETER
NAS1101D HAS BEEN RETIRED, 0512 ALWAYS USED
```

Note: Prior to APAR 2400089, a user could specify a value of zero (0) for the alarm logging table limit which would prevent the table from being created. Since the alarm logging table is an important component for debugging problems, this change means that the table will ALWAYS be generated.

4) Alarm logging logic corrected to fix a counting error with message chains.

Because of an error in the processing of some multiple line alarms (NAS2030I for example), the alarm counts are not correct. This has to do with how the HNAS WTO service routine (XFWTO) processes the WTO. The XFWTO macro and it's service routine have been fixed to process multi-line WTOs correctly so that the alarm logging table reflects the proper alarm count when displayed using the ALARM LOG=? command.

5) DNAS now displays MAINTENANCE/USE ANNIVERSARY DATE IS yyyy/mm/dd even when date is older than current date.

6) Utility changes were also made to the HNAS Authorization Date validation logic to correct an error observed in the trial product expiration date processing.

36) Effective with APAR 2400095, various changes were made to the Authorization Logic in support of the new EOMKEY support.

37) Effective with APAR 2400098, the NASC052W message will no longer be issued when the EXEC LIST command is issued without a command list (*ddname* or (*cmd*1,...,*cmd*n)). EXEC LIST now displays the current setting of the PURGEONERROR|NOPURGEONERROR option in effect. For more information, please see the description of the EXEC= operand on the BUILD definition statement or the EXEC console command.

38) Effective with APAR 2400098, the VARY *sluname* FORCE command will close the TCP/IP socket associated with the SLU even if the SLU is inactive (normally only PVCs can be in this state). <u>Prior to this change, the command would have been rejected if the SLU was inactive</u>. For more information, please see the description of the VARY console command.

39) Effective with APAR 2400098, the NAS9206I message is now withheld until todays date is within 60 days of the EOMDATE. In addition, the message severity is changed from ‘I’ to ‘S’ as todays date gets closer to the EOMDATE. For more information, please see the description of the NAS9206I message in the Messages and Codes documentation.

40) When a PULSE frequency of 60 (for example) is specified:

PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,60)

it is actually treated as 61. To have the PULSE message (NAS0299I) issued once per minute, you would have to specify 59 as the frequency. Effective with APAR 2400102, HNAS has been modified to use the specified frequency value <u>as is</u> rather than adding one (1) to it.

41) The DNAS command is now 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.

n) Additional entries will be provided in this section as APAR Enhancements or Fixes are developed that alter the operation of the code or features.

Migration - V2R3M0

Program Logic changes to V2R3M0 that may affect your environment migration
(On March 6, 2004 some items in this list were reordered in consideration of importance)

1) Beginning with 230, the load module member is now stowed as **HNAS** (**PGM=HNAS**) to accommodate our alternate SMP/E installation support. For HNAS 220 and earlier, the HNAS load module member was stowed as **NASMAIN**.

2) HNAS has been modified to wait until the NAS0001I Host NAS **Initialization Complete,** All Functions Ready state is achieved prior to scheduling HNAS VTAM specific tasks. This prevents potential host operator confusion when VTAM specific HNAS operations were starting prior to the initialization complete message being issued.

3) Changes to **HNAS clear diagnostic** bytes:
   A normal UNBIND from the PLU will result in a clear diagnostic byte of 0 (was 140)
   A non-normal UNBIND from the PLU continues to get a diagnostic byte of 140.
   When HNAS UNBINDs a session the diagnostic byte will be 146. When a session is cleared by a VTAM NOTIFY the diagnostic byte is now 145 (was 140).

4) HNAS has been modified to internally create a **Clear Request** packet when a TCP/IP remote session disconnect condition is detected. This will cause XOT routines to perform cleanup processing which will result in an UNBIND of the LU. This will allow the LU to be released and made available for a subsequent VC connection.

   This logic was actually added to HNAS 220 and propagated to HNAS 230 via APAR 2200048 on 08-06-2003.

5) The **PFXWTO** option will no longer prefix console command output with the NASNAME= operand value. Only HNAS alarm messages (**NAS*innns***) will be prefixed and only when these messages are routed to the operating system master console (SYSCONS) by passing alarm filter (ALRMFLTR=) and alarm limit (ALRMLMTS=) processing. An alarm message that is written to the SYSPRINT log file may or may not contain the NASNAME= prefix depending on whether the alarm passes filtering or limiting. If prefixing is active and an alarm is routed to the SYSCONS, it will also contain the prefix in the SYSPRINT log. However, if prefixing is active and an alarm cannot be routed to the SYSCONS, it will not contain the prefix in the SYSPRINT log.

   The **PFXWTO CONS** start parameter has been added to allow HNAS console output to be prefixed using the existing **NASNAME=** value. This enhancement provides a unique message id that can be used for customer automation processes. This processing requires that the **SHOW CONS ON** option must also be in affect. This processing does not affect alarm output.

   Note: Starting with 220, console command output is, by default, **not routed** to SYSCONS. To override the default and allow console command output to be routed to SYSCONS, specify the SHOWON start parameter or the SHOW ON console command. For 230, specify the SHOWCONS start parameter or the SHOW CONS console command.

Please refer to Enhancement APAR 2300006 (issued on 03-29-2004) for recent changes to this support.

The **PFXWTO *text*** start parameter has been added to allow the *text* value to be appended to the beginning of each alarm WTO in lieu of the NASNAME= operand value. This same text is used to prefix console output if the PFXWTO CONS option is also in affect. Up to 8 non-blank characters may be specified.

Specify **PFXWTO *text* PFXWTO OFF PFXWTO CONS** if you only want console output to be prefixed with the *text* value.

Note: This new feature was introduced into 230 as Enhancement APAR 2300026.

6) **GATE** sessions continue to default to a session start-up delay value of 2 seconds although now there is a new parameter OPTIONS=REQSESSDELAY=*value* which allows users greater flexibility in fine tuning their delay value. The REQSESS delay is used during GATE callout. The delay is imposed after the Call Accept packet has been sent to the CTCP (on the control session LU) and before the REQSESS is issued to ask for a BIND to start the GATE data session. Systems using CFT typically require the delay. Some environments don't require a delay value so they can now disable the timer by coding a value of 0 to eliminate the session start delay.

7) The **TRCPRNT** start parameter and TRCPRNT ON console command no longer controls whether TCP/IP informational messages are written to the SYSPRINT log. This function is now controlled using the **PRNTTCP** start parameter or the PRNT TCP ON console command. The TRCPRNT option has been restored to its original use which is simply to control the logging of trace information to the SYSPRNT log. Please refer to the section titled 'SYSPRINT log filtering' in the V2R3M0 New Features Guide for additional information.

8) The **TRCALL** start parameter and **TRCALL ON** console command no longer include the TRCSUBR or TRCTASK options. If these options are required, they must be specified in addition to TRCALL. Specifying the TRCALL start parameter is the same as specifying TRCBFR, TRCDATA, TRCDBK, TRCDISP, TRCIO, TRCLU MAXDATA, TRCMCH, TRCMCHX, TRCVC MAXDATA. Entering the TRCALL ON console command is the same as entering TRCBFR, TRCDATA, TRCDBK, TRCDISP, TRCIO, TRCLU STRT, TRCMCH, TRCMCHX, and TRCVC STRT. The TRCALL ON|OFF|STRT|STOP console commands no longer alter the state of the TRCLU|VC MAXDATA|MINDATA|NODATA options.

9) Messages containing **clear diagnostic** bytes now contain an additional diagnostic extension byte that provides additional information in cases where there are several reasons for the diagnostic value. This changes the text of the messages but does not change the HNAS message number (NASxxxx).

10) The **severity code** (the ***s-value*** in NAS2*nnn**s*** messages) for multiple line TCP/IP error messages now carry the same severity code for all messages in the sequence. You may need to modify your HNAS ALRTMSG=***filters*** list or host SYSCONS message filtering

mechanism.

11) The **severity code** for NAS1051 and NAS1052 messages has changed from W to E. NAS1051E is issued when HNAS attempts to LOAD a module (for example a USS table) from a library that cannot be opened because the DDNAME was not present in the HNAS start JCL. NAS1052E is issued when the DDNAME for the library is present but the module is not found in the library.

12) Migration item number omitted.

13) HNAS has been modified to generate a **default port value** of 1998 rather than DYNAMIC for a TYPE=XOT REMOTE when an IP address is provided. A default of DYNAMIC is still used when IPADDR=DYNAMIC is specified. This change takes advantage of the shared socket capability that was introduced in 220.

14) For TYPE=XTP|XOT LOCAL and REMOTE definition statements with a valid IP address, a **default port value** of 3065 (XTP) or 1998 (XOT) is forced when a value other than these are coded and the following configuration warning message is generated:

    NAS1211D LOCAL PORT *badval* INVALID, 3065|1998 ASSUMED
    NAS1211D REMOTE PORT *badval* INVALID, 3065|1998 ASSUMED

15) Several HNAS **TCP/IP event alerts** generate a pair of messages. The first message provides the alert message number (***nnn***), severity code (***s-value***) and identifies the resource, for example, SERVER=*ipaddr*(*port*), SOCKID=, PCEID=, NAME=. The second message identifies the event, error condition and appropriate error/reason codes. For HNAS 220, the second message of a pair always used the same alert message number but had an ***s-value*** of ‘**I**’ regardless of the ***s-value*** of the first message. Starting with 230, the entire message identifier for second message will be the same as for the first including the ***s-value***.

    Although the same message identifier is used for both messages of a pair, the alert is only counted once when alarm logging is active. See the ALRMLMTS= operand of the BUILD definition statement and the ALARM console command for more information on alarm logging.

16) The **FASTRUN** process will now propagate VTAM operands that are specified on a TYPE=MXT REMOTE definition statement if that MXT is associated with a SLU in the LUNAME= operand list on a TYPE=SPU REMOTE definition statement. If no MXT is associated with an SLU entry, the VTAM operands are taken from the root TYPE=SPU REMOTE definition statement as in previous releases. In this way, the generated APPL statements for each SLU can have different VTAM operands (for example, MODETAB, DLOGMOD, etc.).

17) The **FASTRUN** process will now allow the name for the AMNF VBUILD statement to be specified using the APPLNAME= operand rather than NASNAME= operand on BUILD definition statement (Enhancement APAR 2300007). If the APPLNAME= operand is omitted, the name will come from the NASNAME= operand as it did in the past. If the NASNAME= operand is also omitted, the VBUILD statement will be produced without a name. This is also true if APPLNAME=NONE is specified.

18) HNAS has been modified to generate a **configuration error message** if GATE=GENERAL and SUBADDRESS=YES are specified without any LLC*i*= operands. The reason is that GATE=GENERAL and SUBADDRESS=YES implies that subaddress values are required but none were provided using the LLC*i*= operands. This condition will terminate HNAS execution after the entire CDF has been scanned. In the past, this condition would not have been discovered until a user attempted to connect which would then have resulted in a Clear.

19) HNAS has been modified to generate a **configuration error message** if GATE=GENERAL, SUBADDRESS=YES and LLC*i*=(*list*) are specified without any SVC*i*= operands. The reason is that GATE=GENERAL, SUBADDRESS=YES and LLC*i*=(*list*) implies that LLC*i* SLU resources are required but none were provided using the SVC*i*= operands. This condition will terminate HNAS execution after the entire CDF has been scanned. In the past, this condition would not have been discovered until a user attempted to connect which would then have resulted in a Clear.

20) HNAS has been modified to generate a **configuration error message** rather than setting default values if the SVC0= operand is omitted or SVC0=NONE is specified and GATE=NO, PAD=NO and SVC3=NONE are specified **or if** GATE=GENERAL, SUBADDRESS=YES and LLC0=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,80,...) are specified. The reason is that these specifications imply that LLC0 (PCNE) SLU resources are required but none were provided using the SVC0= operand. This condition will terminate HNAS execution after the entire CDF has been scanned. In the past, this condition would have caused 64 SLUs to be generated for LLC0 connectivity Now, the number of SLUs and, optionally, their names must be specifically identified.

21) HNAS has been modified to generate a **configuration error message** rather than setting default values if the SVC3= operand is omitted or SVC3=NONE is specified and GATE=NO, PAD=NO and SVC0=NONE are specified **or if** GATE=GENERAL, SUBADDRESS=YES and LLC3=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,83,...) are specified. The reason is that these specifications imply that LLC3 (QLLC) SPU resources are required but none were provided using the SVC3= operand. This condition will terminate HNAS execution after the entire CDF has been scanned. In the past, this condition would have caused 32 SPUs to be reserved for LLC3 connectivity Now, the number of SPUs and, optionally, their names must be specifically identified.

22) HNAS has been modified to generate a **configuration error message** rather than setting default values if the SVC4= operand is omitted or SVC4=NONE is specified and GATE=GENERAL, SUBADDRESS=YES and LLC4=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,<28,...) are specified. The reason is that these specifications imply that LLC4 (GATE) SLU resources are required but none were provided using the SVC4= operand. This condition will terminate HNAS execution after the entire CDF has been scanned. In the past, this condition would have caused 64 SLUs to be generated for LLC4 connectivity Now, the number of SLUs and, optionally, their names must be specifically identified.

23) HNAS has been modified to generate a **configuration error message** rather than setting default values if the SVC5= operand is omitted or SVC5=NONE is specified and GATE=NO and PAD=INTEG|TRANSP are specified **or if** GATE=GENERAL, SUBAD-

DRESS=YES and LLC5=(*list*) are specified **or if** GATE=GENERAL and CTCP=(...,85,...) are specified. The reason is that these specifications imply that LLC5 (PAD) SLU resources are required but none were provided using the SVC5= operand. This condition will terminate HNAS execution after the entire CDF has been scanned. In the past, this condition would have caused 64 SLUs to be generated for LLC5 connectivity Now, the number of SLUs and, optionally, their names must be specifically identified.

24) HNAS has been modified to generate a **configuration error message** rather than setting default values if the CUD0=, SUBD= and CTCP= operands do not all contain the same number of elements. In the past, the number of elements for these operands would have been forced to the highest common value and the following configuration warning message was generated:

NAS1311W REMOTE CUD0 REQUIRES ONE-TO-ONE CTCP MAPPING, FORCED
NAS1311W REMOTE SUBD REQUIRES ONE-TO-ONE CTCP MAPPING, FORCED

Now, the following message will be issued and HNAS will terminate after the entire CDF is scanned:

NAS1311E REMOTE CUD0 AND CTCP COUNTS DO NOT MATCH, REQUIRED
NAS1311E REMOTE SUBD AND CTCP COUNTS DO NOT MATCH, REQUIRED

25) The display ***count*** values for the NAS1709W Configuration message and NAS3798I, NAS4700W, NAS4701W, NAS5700E and NAS5701E Alert messages were changed from Hexadecimal to Decimal to avoid confusion.

26) The **IDLETO=** operand may now be coded for a TYPE=SPU|MXT|MCH|XTP REMOTE definition statement to specify an inactivity timeout value that is used for the HNAS virtual circuit connections associated with the REMOTE definition statement. HNAS monitors each virtual circuit for the absence of input or output. Anytime data is sent or received, HNAS resets the inactivity timeout clock. If no data is sent or received over a virtual circuit after ***minutes*** have elapsed, the virtual circuit call is cleared.

Note: Permanent virtual circuits are exempt from inactivity timeout processing.

You may specify a ***minutes*** value between 0 and 255. A value of 0 causes HNAS to suppress inactivity timeout processing. If a ***minutes*** value is not specified or if the specified value is invalid, a default value will be taken from the **IDLETO=** operand on the **BUILD** definition statement.

Note: If an IDLETO= operand value is specified for an MXT and the MXT is associated with an SLU in the SVC0= or SVC5= operand or an SPU in the SVC3= operand of a **TYPE=MCH REMOTE** definition statement, this value will override the IDLETO= value in affect for the root MCH. In the case of an SPU, the MXT IDLETO= value is used only if the SPU is allocated based on DTE address matching within the SVC3= operand.

Note: If an IDLETO= operand value is specified for an **TYPE=SPU REMOTE** definition statement, this value will used if the SPU is allocated based on IDBLK/IDNUM matching.

27) The **DMAP APAR** command has been modified so that it automatically executes at initialization time *with no delays*. The output of the command is logged in the HNAS SYSPRINT so that maintenance 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 new **DNAS APAR** command. Note that you can still use the DMAP APAR command to display APARs but command output is in module name order rather than APAR ID order. This logic was added to HNAS 230 via APAR 2300004 on 03-19-2004.

28) The **DNAS** command has been modified to display the APAR=*apar-id* of the latest maintenance applied by the customer in addition to the *apar-id* that was incorporated when the HNAS distribution was created. The AUTH=*month-count* value was also added to the DNAS display output to provide the number of months established for the trial period.

    Additionally, a new argument (APAR) has been added to the DNAS command so that the table, created at initialization time by the **DMAP APAR** command, can be displayed. The **DNAS APAR** command will display all APARs on the HNAS system in APAR ID order while the DMAP APAR command displays all maintenance in module name order. This logic was added to HNAS 230 via APAR 2300004 on 03-19-2004.

29) The **INIT=** operand may now be coded for a TYPE=XOT|XTP LOCAL definition statement to specify whether the server connection identified by the LOCAL definition statement should be initially active or inactive. In addition, a retry count and delay time between retires can be specified to control the number of times a stack connection is attempted and the amount of time to delay between connect attempts. If the INIT= operand is omitted, the following is assumed: INIT=(ACTIVE,RETRYLMT=0,**DELAYTIME=5**). Additional information on this support is available in the 230 New Feature Configuration Enhancements section.

    Note: If a TCPIP BIND request fails with ***errno***=30 (see NAS2231W message), the 'linger' timeout is reset so that the next BIND attempt will not end with this error. However, if the INIT= operand is omitted, the **default delay of 5 minutes** will be enforced. This will force HNAS to wait this amount of time before retrying the BIND. To avoid this delay, specify INIT=(ACTIVE,DELAYTIME=0).

30) With APAR 2300037 applied, the way that GATE control sessions and Gate Fast Connect data sessions are activated has been changed as follows: If a REQSESS VTAM operation fails the NAS3702W message is issued to indicate that HNAS cannot set up the GATE control fast connect data session. Before 2300037, HNAS would wait for a BIND from the PLU. With the APAR applied HNAS will retry the REQSESS based on the OPTIONS=MCHTMR=xx value specified in the CDF (xx default=60 seconds). If HNAS is left active when GATE CTCPs are inactive the number of NAS3702W messages will increase.

31) 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 (previous specified on the BUILD ALRMFLTR=) so that they could add new message ID's via the console.

With APAR 2300045 applied, 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 IDs in order to add just one entry to the end of the list. For example:

If `BUILD ALRMFLTR=(ALLOW,NAS2***I(P))` was specified in the CDF and if `ALARM FILTER=(NAS3701W(P))` is issued as a console command, the `ALARM FILTER=?` command would produce the following display:

```
ALRMFLTR=ALLOW
          NAS2***I(P)
          NAS3701W(P)
```

If you wish to clear the `ALRMFLTR=` array before entering new values, enter `ALARM FILTER=*` to clear the entries followed by `ALARM FILTER=(`*`disp`*`,`*`list`*`)` for the new filter entries.

32) The non-SMP/E **hnas.ZIP** and SMP/E **lns.ZIP** distribution **file names** were **expanded** to provide information regarding the HNAS release and maintenance level of the product provided with-in the zip achieve file. The support is effective June 21, 2004.

33) The **DNAS** command has been modified to display a list of missing *apar-ids* if the 'latest' *apar-id* is different than the *apar-id* that was incorporated when the HNAS distribution was created. The missing *apar-id* logic is only executed when DNAS is entered with no arguments. This new feature generates a minimum of 2 additional lines (line 6-n depending upon maintenance manually applied) to the DNAS display output.

Please refer to the 'DNAS Command - Display HNAS Information' section of the Console Subsystem Operations Guide for additional information.

Note: This new feature was introduced into 230 as APAR 2300062.

34) **CustomUserMods** are now provided in the HNASMACX and/or HNASOBJX distribution files for customers that have custom enhancements carried forward for refresh and/or upgrade distributions. This was done to separate their product distribution custom enhancements from the standard distribution source (macro) and object members to improve customer enhancement documentation and maintenance support services.

Note: This new feature was introduced into 230 as APAR 2300069.

35) Effective with APAR 2300017 (04-13-2004), **TAP operation** is slightly different than it was prior to this APAR. Older TAP logic allowed the router to timeout an active socket connection because the TAP socket was opened at the beginning of the TAP= timeout interval but the TAP packet was not transmitted until the TAP= timeout expired. The router expects to see data within a 120 second time period after a socket connection is established. If the TAP= timeout value is greater than 120 seconds, no data will be received by the router before its data timeout expires. In this case, the router will close the socket connection. This will prevent the complete TAP Keep Alive sequence from completing correctly. While the socket closure indicates that the router is active, it does not indicate whether an addressed MCH is active. The new TAP logic delays opening of the socket

until the TAP= timeout expires so that the socket opening and TAP packet transmission are done at approximately the same time. This prevents the router from timing out the socket connection. This, in turn, will allow a TAP packet that addresses a specific X.25 MCH on the router to reach its destination and the response from the MCH (either Clear or Call Confirm if the TAP packet is a Call Request) to get back to HNAS. Clear cause and diagnostic codes help determine the MCH state.

Note: This new feature was introduced into 230 as APAR 2300017. This migration APAR item was erroneously overlooked and not added to the 230 migration documentation section until 09-02-2004.

36) The HNAS configuration processor has been modified to set **PVC=NONE** as the default when the PVC= operand is omitted. A **NAS1301D** informational message will now be produced rather than a **NAS1301W** warning message. This will eliminate the requirement of environments without PVC resources having to code PVC=NONE to eliminate the CC-4 condition codes previously set by the warning message.

Note: This logic change was introduced into 230 under APAR 2300077.

37) **Clarification for the SHOW*type*** start parameters and their equivalent SHOW *type* console commands. The SHOWERR, SHOWON, SHOWOFF and SHOWCONS start parameters provide the same function as the SHOW ERR, SHOW ON, SHOW OFF and SHOW CONS ON console commands. The SHOWERR, SHOWON and SHOWOFF functions are mutually exclusive. The SHOWCONS function is independent.

SHOWERR allows only error alarm messages to be displayed at the SYSCONS.

SHOWON allows error and informational alarm messages to be displayed at the SYSCONS.

SHOWOFF inhibits all alarm messages from being displayed at the SYSCONS.

When SHOWERR or SHOWON are in affect, alarm messages are also subject to filtering and limiting which may prevent them from being displayed at the SYSCONS.

The SHOWCONS start parameter and SHOW CONS ON console command are not related to the other SHOW*type* start parameters or SHOW *type* console commands, respectively. SHOWCONS only governs whether console command output will be displayed at the SYSCONS.

HNAS activates with SHOWERR and SHOWCONS OFF in affect unless specifically overridden by the user. This means that only error alarm messages will be displayed at the SYSCONS and console command output will be restricted to SYSPRINT.

Note: Regardless of the SHOW*type* setting, all alarm messages are logged in SYSPRINT unless PURGE is specified as the default ALRMFLTR disposition (first suboperand) or the alarm message is specifically filtered using the NAS*xxxxx*(P) option. SYSPRINT logging is always active when HNAS is started but may be inhibited for all alarm and console command output using the PRNT OFF console command.

38) The **DNAS** command has been modified to minimize the confusion regarding missing maintenance. 'MAINTENANCE STATUS' is displayed instead of 'MISSING MAINTENANCE' in the header text and 'NOT INSTALLED' is displayed instead of 'MISSING' under MODULE when an APAR is not applied. When all APARs are installed (up to the most recent APAR ID), 'ALL MAINTENANCE ON ...' will be displayed instead of 'NOTHING MISSING'. Hopefully, these changes will eliminate customer concerns and confusion when some maintenance is not installed.

<u>OLD display:</u>

```
APARID   MODULE  (MISSING MAINTENANCE)
230xxxx  MISSING
:
230xxxx  MISSING
 -or-
APARID   MODULE  (MISSING MAINTENANCE)
NOTHING MISSING
```

<u>NEW display:</u>

```
APARID   MODULE  (MAINTENANCE STATUS)
230xxxx  NOT INSTALLED
:
230xxxx  NOT INSTALLED
 -or-
APARID    MODULE  (MAINTENANCE STATUS)
ALL MAINTENANCE ON THROUGH MOST RECENT APAR xxxxxxx
```

Note: This logic change was introduced into 230 under APAR 2300089.

39) A default value for the **CUD=** operand is no longer set for a TYPE=MCH|XTP REMOTE definition statement. This change was implemented by APAR 2300095 (January 2005) so that the correct default CUD can be set for PCNE and PAD calls. Prior to this APAR, a default CUD of 01000000 was set for both PCNE and PAD. This forced the remote DTE to provide PAD services for a PCNE call. By not setting a default for the CUD= operand, the correct default CUD can be set for PCNE and PAD VCs when the outbound Call Request packet is built.

If no ***cud*i** suboperand is specified and the CUD= operand is also omitted, a default value of C0000000 will be used for PCNE (LLC0) and a default value of 01000000 will be used for PAD (LLC5).

40) A CR/LF (0D/25) prefix and suffix are no longer added to a **USSMSG** sent to a QLLC SLU. This additional output data caused problems on subsequent LOGON input from the terminal operator. The input was being truncated so that the LOGON was rejected during **USSTAB** processing and caused USSMSG2 ('COMMAND UNRECOGNIZED') to be transmitted.

Note: This change was implemented by APAR 2300096 (January, 2005).

41) The **HNASXEQ** job in the *qqqq*.HNASCNTL dataset that is shipped with HNAS now contains additional SYSPRINT DD statements (SYSPRNT2 and SYSPRNT3). These are provided as an example of how multiple SYSPRINT files can be specified for use by the PRNT CLSOPN console command. Note that the original HNASXEQ job with a single SYSPRNT DD statement is also present but has been renamed to HNASXEQO. This change is included in distributions created after 2005-01-17.

42) The **TRCALL ON|OFF** console command no longer operates on PCEs only. The PCE trace function has been removed form the ON|OFF processing. ON and OFF are now treated the same as STRT and STOP, respectively. The PCE trace function is now controlled by the **TRCPCE ON|OFF|ALON|ALLOFF** console command. Note that TRCPCE is a console command only. TRCPCE provides the function as TRCBFR, TRCDATA, TRCDISP and TRCIO together.

Note: This logic change was introduced into 230 under APAR 2300110.

43) The **TRCBFR**, **TRCDATA**, **TRCDISP** and **TRCIO** console command processors have been modified to accept the **ALLON|ALLOFF** arguments. These arguments provide for global PCE tracing while the existing ON|OFF arguments provide local PCE tracing using the RNM= or ID= modifiers.

Note: This logic change was introduced into 230 under APAR 2300110.

44) Some of the alarm messages that were generated by the **TRCALL**, **TRCTRAP** and **TRCPRNT** console command processors have been modified to display the name of the console issuing the command. Where the console name is now displayed, the text 'COMMAND' used to be displayed. The following alarm messages were changed:

For TRCPRNT ON

NAS0210W SYSPRINT TRACE LOGGING ENABLED BY *consname*,
MORE CPU CYCLES REQUIRED

For TRCPRNT OFF

NAS0211I SYSPRINT TRACE LOGGING DISABLED BY *consname*

For TRCALL SUSP or TRCTRAP SUSP

NAS0050A TRACING SUSPENDED BY *consname*

For TRCALL RSME or TRCTRAP RSME

NAS0060W TRACING RESUMED BY *consname*

For TRCTRAP SNAP

NAS0080A SNAPSHOT DUMP TAKEN BY *consname*

For TRCTRAP RSMESNAP

NAS0080W SNAPSHOT DUMPING RESUMED BY *consname*

Note: This change was made so that a SYSCONS console operator will know if the trace state has been changed by a remote console operator. For the local SYSCONS, *consname*=WAPCECON. For a remote console, *consname*=RCON*xxxx* where *xxxx* is the number of the REMOTE console PCE.

Note: This logic change was introduced into 230 under APAR 2300110.

45) From traces collected by a customer, it appears that an SPU can return a **QXID** request as a response to a QXID request that HNAS sends. HNAS ignores SPU QXID request and a timeout occurs even though the SPU QXID request carries a valid IDBLK/IDNUM value. The timeout causes following alarm message to be issued:

NAS8191W CLIENT=002.241.202.250(01178) SOCKID=0001
PCEID=0032 NAME=XOTCNOT1

NAS8191W XID TIMEOUT FOR PU MTA31PUR

HNAS retries the QXID request 3 times and eventually clears the call with CAUSE/DIAG=000/089.

Starting with APAR 2300116, a QXID request from an SPU that is received as a response to a QXID request that HNAS sends will be treated as a QXID response if the SPU is configured as the primary (OPTIONS=PRI) or as a peer link station (OPTIONS=PEER). Only when the SPU is configured as the secondary (OPTIONS=SEC) will the QXID request will be treated as a request.

Note: This change was implemented by APAR 2300116 (March, 2005).

46) From traces collected by a customer, it appears that HNAS is receiving **INIT-SELF** requests for SLUs that are still waiting for their ACTLU response. HNAS should reject these requests but does not. The INIT-SELF (or any other request in ACTLU pending state) violates the rules for Immediate Response Mode protocol.

Starting with APAR 2300116, HNAS will now reject any request received from an SLU before the ACTLU completes with 200D error sense (Response Owed Before Sending Request).

Note: This change was implemented by APAR 2300116 (March, 2005).

47) The VARY console command was modified to allow a single socket on a router to be manipulated using the ID= command modifier. Prior to this change, all sockets on a router were manipulated because the command resolution was limited to a REMOTE level using the RNM=*rmtname* modifier. While the RNM= modifier can still be used to activate or deactivate a router, the ID= modifier can now be used to activate or deactivate a single socket or a group of sockets on a router. As part of this change, logic was also added to ask the operator to confirm his VARY request in order to prevent an inadvertent change that could disrupt a production environment. In the past the VARY command was

executed immediately with the given parameters as soon as the enter key was pressed. Now, the VARY command will echo back the parameters that were entered and require a YES or NO confirmation response before actually executing the command.

Step 1: **RNM=XOTCLNT1 DPCE** <- set the router name and display its sockets

```
 PID  NAME     NASOPT TYPE TYPQ PROT STAT SUBST IPADDR ...
 0008 XOTCLNT1         TCP  RMT  XOT IDLE T     010.117...
 0009 XOTCLNT1         TCP  RMT  XOT IDLE       010.117...
 000A XOTCLNT1         TCP  RMT  XOT IDLE       010.117...
 000B XOTCLNT1         TCP  RMT  XOT IDLE       010.117...
 000C XOTCLNT1         TCP  RMT  XOT IDLE       010.117...
```

Step 2: **V OFF** <- deactivate entire router (RNM=XOTCLNT1 still set)
Note that TAP socket (T in SUBST column) is not affected.

```
VARY PARMS: ACTION=OFF TYPE=RMT RNM=XOTCLNT1
    ENTER: N=ABORT, Y=EXECUTE
```

Step 3: **V RNM= ID-9 ON** <- reset the router name (temporarily) and activate the first (non-TAP) socket

```
VARY PARMS: ACTION=ON  TYPE=N/A ID=0009-0009
    ENTER: N=ABORT, Y=EXECUTE
```

Step 4: **DPCE** <- display router sockets (RNM=XOTCLNT1 still set)

```
 PID  NAME     NASOPT TYPE TYPQ PROT STAT SUBST IPADDR ...
 0008 XOTCLNT1         TCP  RMT  XOT IDLE T     010.117...
 0009 XOTCLNT1         TCP  RMT  XOT IDLE       010.117...
 000A XOTCLNT1         TCP  RMT  XOT CLSD       010.117...
 000B XOTCLNT1         TCP  RMT  XOT CLSD       010.117...
 000C XOTCLNT1         TCP  RMT  XOT CLSD       010.117...
```

Note: This change was implemented by APAR 2300123 (March, 2005).

48) HNAS will now issue a Clear **diagnostic code 211 x'D3**' instead of a Reset packet sequence due to improvements in the SIGNAL processing. Please refer to APAR 2300140 for additional information.

Note: This change was implemented by APAR 2300140 (June, 2005).

49) The default value for the DELAYTIME= suboperand of the INIT= operand for a LOCAL definition statement has been changed from 5-minutes to 1-minute. If a BIND the failure occurs, the smaller DELAYTIME= default should make BIND error recovery operate more quickly. In addition, HNAS now resets the TCPIP ‘linger timeout’ before the initial BIND is issued instead of after the first failure. This should minimize the probability of receiving the EADDRINUSE error condition (***errno***=30 in the NAS2231W BIND FAILURE message) and further speed up HNAS initialization processing. For more information on the EADDRINUSE error condition, please see the description of the INIT= operand for the

LOCAL definition statement in Chapter 4 of this document.

Note: This change was implemented by APAR 2300150 (August 9, 2005).

50) Changes to HNAS **PVC reset packet** processing were implemented to improve network PVC VC (virtual circuit) status reporting to the Host PLU.

When a reset is received for an LLC0 or LLC5 PVC with a cause/diagnostic of 029/115 (network/link out of order) or 001/000 (out of order) then the PLU will be notified of the error by a NOTIFY request generated when the HNAS ACB for the session is closed. Other resets continue to be reported with SIGNAL PIUs sent to the PLU.

When a reset request other than 015/000 or 000/000 (network/device operational) is received for an LLC3 session all LUs associated with the VC's PU are be taken down (PLU will receive NOTIFY). This stronger treatment is provided for LLC3 because when a RESET is received it is not possible to tell which LUs are affected.

Note: This change was implemented by APAR 2300151 (August 16, 2005).

51) Changes to the HNAS **PVC Setup Status Codes** were implemented to improve Cisco router PVC setup retry logic. Following list contains the status code reassignments with notations, as required:

New status codes used by HNAS to reject inbound SETUP packets. Codes greater than X'0F' cause the router to never retry the setup. The new codes allow the router to retry the setup every 5 minutes, as appropriate.

X'**13**' = no such destination interface is now X'**0C**'. This code is used when the router has an incorrect HNAS MCH name.

X'**14**' = destination interface not up is now X'**08**'. This code is used when a SETUP is received when HNAS is coming down.

X'**16**' = no such destination PVC is now X'**0D**'. This code is used when a SETUP is received for an HNAS MCH but the setup's LCN has not been defined in HNAS.

X'**19**' = can't support flow control values, is now X'**09**'. This code is used when a SETUP is received specifying a window size > 7 or a packet size > 4096.

X'**1A**' = PVC setup protocol error, is now X'**0E**'. This code is used when a SETUP is received for a PVC that already has a session.

Please refer to RFC1613 and the HNAS Messages and Codes PVC Status Field Sense Codes section for additional information.

Note: This change was implemented by APAR 2300151 (August 16, 2005).

52) The VARY console command was modified to accept the new FORCE option so that the OFF (or INACT) function for LOCAL and REMOTE resources operates in a similar fashion for the purpose of closing active remote client sockets. Prior to enhancement APAR

2300156, V LCL OFF closed the LOCAL socket and marked the LOCAL offline but did not close any associated remote CLIENT sockets. Conversely, V RMT OFF closed all active remote client sockets and marked the REMOTE offline. This lack of symmetry is corrected by using the FORCE adverb in conjunction with the OFF verb as follows:

V LCL OFF will close the LOCAL socket and mark the LOCAL offline (no change).

V LCL OFF FORCE will close the LOCAL socket, mark the LOCAL offline and also close any active remote client sockets for which the LOCAL is HOME (new functionality).

V RMT OFF will mark the REMOTE offline but will not close any remote client sockets (changed functionality).

V RMT OFF FORCE will close all active remote client sockets and mark the REMOTE offline (FORCE is required to perform the old functionality).

Note: This change was implemented by APAR 2300156 (September 29, 2005).

53) The HNAS PADPARM=value option now utilizes a Set instead of a Set & Read for X.25 Pad resources to better simulate the operation that NPSI performs. Please refer to the HNAS PADPARM= parameter in Chapter 4 of the HNASBook for additional information.

Note: This change was implemented by APAR 2300159 (October 13, 2005).

54) The NASC300E console command error message will now 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.

Console commands that require RNM= have always been rejected with the following message issued when RNM= was omitted:

NASC300E RNM= OMITTED, REQUIRED

Note: This logic change was introduced into 230 under APAR 2300161.

55) Suffix modifiers (right side modifiers) that are entered for commands that do not use modifiers (e.g., DPARM) will now be treated as command arguments. Prior to APAR 2300161, right side modifiers were processed even though they were not used by the command that preceded them. This precluded their use as a command argument. After APAR 2300161, right side modifiers for commands that do not use modifiers will be passed to the command processor so that they can be processed as an argument. For example, if DPARM RNM= is entered, RNM= will now be decoded by the DPARM command processor and will cause the value of the RNM= modifier to be displayed.

Note: This logic change was introduced into 230 under APAR 2300161.

56) The default HOST= operand has been changed from OS390 to ZOS. This affects the non-SMP HNASRCV installation job (NASMAIN and NASTCP assemblies) but is otherwise transparent to HNAS operation. For SMP/E installation, the NASMAIN and NAS-

TCP assemblies should now always end with CC=0 regardless of whether HOST=ZOS or HOST=OS390 is specified. Note that the DNAS display has also been modified to display the actual HOST= operand (either ZOS or OS390) from the NASMAIN assembly rather than simply HOST=OS390|ZOS.

Note: This logic change was introduced into 230 under APAR 2300162.

57) On November 1, 2005 changes were made to the SMP/E distribution mechanism to tailor the process based upon recommendations from our Business Partners and direct SMP/E customers;

1) A single LNS.*aparid_date_cust#_custid*.ZIP edistribution level file is now provided instead of an edistribution level subdirectory containing Individual distribution files;
2) The FTPGET JOB in SLNSCNTL (used to automate fetching of the individual files) is no longer generated;
3) The HNAS distribution date and APAR level is now included SMP/E maintenance ++FUNCTION, DESC() area;
4) The SMP/E edistribution file packaging is now similar to non-SMP/E edistributions.

58) The DPARM EXEC command will now display ALLON|ALLOFF instead of ON|OFF for TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCMCH, TRCMCHX and TRCVC to more properly describe *global* trace states.

Note: This logic change was introduced into 230 under APAR 2300165.

59) The PRNTQLLC start parameter and PRNT QLLC ON|OFF console command have been added to allow QLLC VC informational alarm message (NAS8*xxx*I) logging to be controlled by its own option rather than with the PRNTVC start parameter or PRNT VC ON|OFF console command as in the past.

Note: This logic change was introduced into 230 under APAR 2300167.

60) The **DNAS** command has been modified to display the running operating system type and version level as well as the time and date when HNAS was started. In addition, some existing DNAS display items have been relocated.

OLD display:

```
VERSION=V2R3M0 HOST=ZOS    ASMDATE=10/28/05 DIST=NON-SMP
CUSTMAC=COMM1.TEST.HNASMAC
CUSTOBJ=COMM1.TEST.HNASOBJ
CREATED AT 14:33:16 ON 12/13/2005
CREATED WITH MAINTENANCE THROUGH APAR 2300167
MOST RECENT MAINTENANCE APPLIED IS APAR 2300167
SHIPID=9999999999999999 AUTH=00

APARID   MODULE  (MAINTENANCE STATUS)
ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2300167
```

NEW display:

```
VERSION=V2R3M0 DIST=NON-SMP ASMDATE=12/16/05 ASMHOST=ZOS <-chgd
RUNNING UNDER z/OS 01.04.00                          <- new line
STARTED AT 10:54:50 ON 12/16/2005                    <- new line
CREATED AT 10:11:31 ON 12/16/2005
CREATED WITH MAINTENANCE THROUGH APAR 2300168
MOST RECENT MAINTENANCE APPLIED IS APAR 2300168
SHIPID=9999999999999999 AUTH=00
CUSTMAC=COMM1.TEST.HNASMAC                           <- moved
CUSTOBJ=COMM1.TEST.HNASOBJ                           <- moved
                                                     <- new line
APARID   MODULE  (MAINTENANCE STATUS)
ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2300168
```

Note: In this example the product was generated and installed on the same date. The 'CREATED AT' date and ASMDATE date are normally the same because the CONSDNAS module is assembled as part of the distribution creating job.

Note: This logic change was introduced into 230 under APAR 2300168.

61) **USEMDFY** is now a default start parameter. This means that the system MODIFY command interface will now be the default method for providing input to the HNAS console subsystem. In the past, WTOR was the default method for entering console input. With this new default, USEMDFY can now be removed from the startup parameter list. To restore WTOR as the console input method, you need to specify USEMDFY OFF or USEWTOR {ON}. Note that USEWTOR is a new start parameter that was added for this change. USEMDFY ON|OFF is treated the same as USEWTOR OFF|ON, respectively.

Note: This logic change was introduced into 230 under APAR 2300169.

62) **TAP operation** response timer is now equal to the TAPping timer interval. Prior to this APAR the response time was 1/2 of the tap timer interval. Under the old logic, when TAP=10 was specified a response timer of 5 was enforced which was too short for some router networks.

In addition, the configuration process has been changed to inhibit default messages for TAP parameters like CUD=, DCEADDR=, DTEADDR= and FAC= when TAP=0 is specified. These default messages were inhibited when TAP= was omitted which sets 0 by default. Specifying TAP=0 or allowing it to default when TAP= is omitted should be treated the same way, that is, tapping is not to be activated and hence, these TAP parameters are not required. Note also that if these TAP parameters are specified for a REMOTE for which tapping is inhibited, a informational message will be displayed indicating that the specified parameters will be ignored but will be saved in the event that tapping is started via a console command.

Note: This logic change was introduced into 230 under APAR 2300173.

63) The **TRCMCH INI** command has been retired. This command set a flag that was never tested. When HNAS was first written, it was assumed that MCH initialization processing would be traced based on this command but subsequently this trace activity became unnecessary.

Note: This logic change was introduced into 230 under APAR 2300174.

64) **NAS3705W** alert message sense code conditions of 0813xxxx or 0814xxxxs are now reported under an informational **NAS3705I** alert message. These sense values indicate bracket race condition that are recoverable.

Note: This logic change was introduced into 230 under APAR 2300177.

65) **HNASRCV** job was modified to remove the sample (NOP) ZAP step that is rarely, if ever utilized in the product distribution installation or maintenance (also affects the **HNASMNT** job which is created from the HNASRCV job). If a ZAP (patch) is required for a pre-APAR emergency problem fix JCL can be provided when the temporary fix is delivered.

APAR PTF's are no longer distributed with ZAP (patches), we now only employ source (macro) and object PTF's.

Another reason for the removal was because some systems personal responsible for HNAS installation are not authorized to use the super ZAP program (AMASPZAP) which caused unnecessary confusion.

Note: This change was made for distributions created after February 2, 2006.

66) **NAS0910I** alert message '3 BELLS AND ALL IS WELL' is now written to SYSCONS as well as SYSPRINT. Routing is no longer controlled by SHOWERR. This message is issued once per day at the midnight crossover. Writing the message to SYSCONS now allows it to also be routed to NETVIEW which some customers wanted.

Note: This logic change was introduced into 230 under APAR 2300182.

67) **NAS25xxM** monitor messages can no longer be filtered via ALRMFLTR=(NAS25xxI(P)) on the BUILD definition statement. This 'loophole' that allowed monitor messages to be filtered (M was being treated as I internally) has been closed. Like their trace message counterparts (for example, NAS7718T), monitor messages must never be filtered from SYSPRINT. If they are not wanted, MON TAP ALLOFF should be entered to terminate TAP monitoring.

Note: This logic change was introduced into 230 under APAR 2300183.

68) **NAS4707W** LU lu-nm GENERATING ERR/INFO PACKET FOR CTCP... alert message type for "CMD RCV'D 17 HNAS ERROR CODE 1, 2 or 3" condition will no longer be generated. We have been advised that NPSI silently discards a clear confirm from the CTCP. HNAS has been modified to do the same.

Note: This logic change was introduced into 230 under APAR 2300185.

69) The **TRCLU**, **TRCLUQ**, **TRCMCH**, **TRCMCHX**, **TRCPCE, TRCVC** and **TRCVCQ** command processors have been modified to reject a request when RNM=, LUNM= and ID= are all omitted. The following error message is also issued:

```
NASC100E ID= OMITTED, REQUIRED WHEN LNM=, RNM= OR
                     LUNM= IS NOT SET
```

If ID=0 is in effect when RNM= and LUNM= are not set, these commands operate on all PCEs associated with the corresponding resources as they always have. For additional information on the command modifier hierarchy, refer to the Console Subsystem documentation.

Note: This logic change was introduced into 230 under APAR 2300188.

70) HNAS has been modified to execute the **DNAS** command when HNAS is started, unconditionally. DNAS no longer has to be specified in the CONCMDQ= operand and will no longer be a default queued command if CONCMDQ= is omitted. Unlike the DNAS command in the CONCMDQ= list which is executed after the NAS0001I INITIALIZATION COMPLETE is issued, the new DNAS logic executes as soon as the CDF scan completes.

The **DMAP APAR** command is also executed after the CDF scan is complete (before DNAS) in order to populate the APAR table and find the highest APAR number on the system which DNAS displays. Starting with this APAR, the DNAS APAR command that is executed at startup will no longer write output to SYSPRINT. If you wish to see DMAP APAR output, you can enter the command manually or specify it in the CONCMDQ= operand.

Note: This logic change was introduced into 230 under APAR 2300192.

71) The **TRCMCH ALLON|ALLOFF** console commands no longer manipulate the ICR, OCR, ICLR and OCLR arguments as was erroneously done in the past. Now these arguments can only be manipulated using individual console commands (e.g., TRCMCH ICR, TRCMCH OCR,..., etc.). The Chapter 2 Start Parameter section in older versions of the 230 HNASBook documentation manual (prior to May 10, 2006) incorrectly indicated that the TRCMCH ALLON|ALLOFF start parameters enable or disable the individual ICR, OCR, ICLR and OCLR arguments.

In a future enhancement, the TRCMCH arguments will be supported via start parameters (PARM=) in addition to existing console command input or CONCMDQ= methods. This change is primarily being implemented so that TRCMCH ALLON|ALLOFF start parameter and console commands are consistent.

Note: This logic change was planned for V2R3M0 but has deferred until V2R4M0.

72) **NAS7708W** alert message was altered so that the Initiator PVC number and Responder PVC number are now included in the message.

NAS7708W XOT PVC SETUP INIT=SERIAL*mch-name* PVC number
RESP=*SERIALn/n* PVC *number*

Note: This logic change was introduced into 230 under APAR 2300196.

73) PVCs are no longer marked with the inoperative bit since there are cases where HNAS is not notified that the Serial Interface PVC is operational unless data is sent or received. This logic was initially implemententent via APAR 2300151 in an effort to avoid PLU data delivery attempts to a remote PVC that isn't available.

Note: This logic change was introduced into 230 under APAR 2300196.

74) **NAS7718T** *ipaddr*(*port*) PVCSETUP TO MCH *mch-name* trace record is no longer erroneously generated by default in SYSPRINT when TRCPRNT is enabled. The record is now only generated when enabled via the TRCMCH ICR console command.

Note: This logic change was introduced into 230 under APAR 2300196.

This page left blank intentionally.

Migration - V2R2M0

Program Logic changes to V2R2M0 that may affect your environment migration

1) HNAS **now requires an authorization file (NASAUTH)** in order for it to operate. The authorization file is specified by the **//AUTH DD** statement in the HNAS start job. For example:

   //AUTH DD DSN=*qqqq*.HNASMAC(NASAUTH),DISP=SHR

   As part of the distribution build process, HNAS identification and authorization files (NASIDENT and NASAUTH) are created and stowed in the HNASMAC macro library. The DNAS console command module (CONSDNAS) is then assembled as part of the HNAS distribution process to pickup the current NASIDENT copy code.

   For users undergoing HNAS trials, a NASAUTH file is provided with a 90 day expiration date. In this case, HNAS will issue an **alert message** everyday at midnight indicating how many days are left before the authorization expires. If the authorization expires while HNAS is running, it will continue to run. However, if HNAS is stopped, it will not be able to be restarted. In addition to the expiration date, the NASAUTH file also contains a SHIPID which is unique to each customer. The SHIPID in the NASAUTH file is compared against the DNASID that can be displayed using the DNAS console command. If the SHIPID does not match the DNASID, HNAS will not be able to start. HNAS will treat the SHIPID/ DNASID mismatch as an authorization failure. A permanent authorization file will be provided following the trial tests when HNAS is accepted by the customer.

   If the //AUTH DD statement is not specified or if the specified authorization file has expired, HNAS will terminate (ABEND) at startup time. An HNAS image that is currently running when the authorization expires will continue to run until stopped by the console operator. However, if HNAS is stopped, it will not be allowed to restart until a permanent authorization file is provided. Customers who have completed their trial tests and have purchased HNAS will be given a permanent authorization file that will never expire.

2) Changes were made to the HNAS 220 **edistribution** for ***.STR** stream and ***.ZIP** file formats. The ***.STR** stream files now contain OS unloaded PDS libraries generated using the TSO XMIT command and now loaded using the TSO RECEIVE command. The IKJEFT01 (batch TSO command processor) is primarily used to invoke the RECEIVE process at the customer site. For ***.ZIP** distributions all product edistribution stream files are now contained in a single ZIP file. We no longer distribute the HNASMAC or HNASOBJ installation files as binary sequential LRECL=80 *.STR stream files in our standard edistribution. Please refer to Chapter 2 (Installation and Generation) for additional information.

3) We no longer distribute HNAS physical media on **3480 cartridge tapes.** **CD** is the primary physical distribution media available although DAT tape can be special ordered, if required. eDistribution is the primary distribution method available via **FTP**, **e-mail attachment** or **IND$FILE** transfer services.

4) **Shared Socket Support (REMOTE TYPE=XOT IPADDR=*a.b.c.d*,PORT=1998)**.
A single TYPE=XOT REMOTE definition statement with a **fixed IP address** and **PORT=1998** is all that is required to create an ***shared socket pool*** (inbound and outbound) for a specific router. The VCLMT= operand value specifies the number of sockets in the pool. If multiple REMOTE definitions with the same fixed IP address and PORT=1998 are provided, the sum of their VCLMT= operand values will be used to build the shared socket pool for the target router. A socket from the pool is allocated to an activating session on a first come, first served basis. The same socket can be used for an inbound connection at one time then an outbound connection at another time.

If you are planning to use all three types of socket pools (**dynamic**, **inbound** and **shared**), they should be specified in this order within the CDF. This will allow sockets to be allocated from the ***dynamic socket pool*** before any are allocated from the ***inbound socket pool*** and from the ***inbound socket pool*** before any are allocated from the ***shared socket pool***. This ensures that more sockets from the ***shared socket pool*** remain available for outbound connections. Please refer to Chapter 4 operand **IPADDR=** for a complete description of this support.

5) **Callin LLC0 (SVC0=) and LLC5 (SVC5=) DTE Address Filtering** for **TYPE=MCH|XTP** REMOTEs SLUname association. HNAS now supports filtering of DTE addressed based on the number of digits coded on the SVC0= and SVC5= dteaddri suboperands. In earlier releases the inbound DTE address in the call request packet had to match the SVC0|5 dteaddri value exactly for LU selection to occur. With the new logic if SVC0=(...,LU1/1234I,...) is coded then LU1 is selected by a call request packet with a calling DTE address of 1234,12345, 123456 etc. (with the previous logic, only 1234 would select LU1). Please refer to Chapter 4 for revised information on SVC0 and SVC5 coding rules.

6) **Callin with GATE=GENERAL,CONNECT=NO,SUBADDR=YES** logic was corrected to set the LLC using CUD0 then subaddress. Earlier HNAS levels erroneously sets the LLC using the subaddress before examining the CUD0 value. <Ref: 2110018 09-05-2002>

This logic was backed out. Please refer to APAR Ref: 2200002 11-21-2002.

7) **Callin SLU/PLU fixed connection support for PCNE (LLC0) and PAD (LLC5).** Redesigned 220 logic now supports coding of an MCH **APPLNAME=** index entry for direct mapping instead of the previous 211 requirement for a SYSL=DATA=*char* system select value. Please refer to Chapter 4 for revised information on **SVC0** and **SVC5** coding rules. The following table depicts the differences in 211 and the new 220 coding requirements:

Old Method Under V2R1M1	New Method Under V2R2M0

`SVC0=(2,`
`MCH10001/123456IA,`
`MCH10002/X001540IB)`
`SYSL=(DATA=A/0,DATA=B/1)`
`APPLNAME=(TSO,CICS)`
	`SVC0=(2,`
	`MCH10001/123456I0,`
	`MCH10002/X001540I1)`
	`APPLNAME=(TSO,CICS)`

8) **Callin Default PLU assignment via SVC0=/SVC5= APPLNAME= association** is now supported in HNAS. For **LLC0 and LLC5 inbound calls**, after HNAS sets the LLC TYPE and locates an LU for the call, a PLUname for the session must be determined. If the name is not provided by the **SVC0/5=** operand (SVC5=(...,LU1/I2,...) where 2 is an index in APPLNAME=) or by SYSL= then the first **APPLNAME=** entry is used as a default. See sample coding in item 7 above. This default is also taken if **SYSL=** is omitted (**SYSL= was required** in previous releases and is **now optional**). Please refer to Chapter 4 for revised information on SVC0= and SVC5= coding rules.

9) **TAP=***value* can now only be coded on the primary TYPE=XOT REMOTE for each unique IPADDR=. This is really a cosmetic change because the TAP=value on the secondary remote's was never used for tapping although the CDF configuration decode made it seem so. This parameter change will now eliminate confusion regarding which remote resource represent the actual tapping value.

   TAP keep alive logic was redesigned and now schedules an XOT Call Request instead of an XOT Clear Request to ensure that a protocol level response (in this case a Clear Request) is always received from the router's XOT services.

   TAP= logic can now perform protocol level tapping with XOT Call or XOT Clear packet. See new parameter OPTION=TAPWITHCLR that was added on 2003-08-27 via APARs 2200048 and 2200052.

10) **RTEIN= (TYPE=MCH)** and **RTEOUT= (TYPE=XOT)** LOCAL **rmtnamei** entries must now correspond to valid TYPE=XOT REMOTE names or a cc-8 configuration error message will be generated and HNAS activation will terminate at the end of the CDF scan process. In previous HNAS levels a cc-4 configuration warning messages was generated which allowed the activation process to continue.

11) **LLC0=, LLC4=, LLC5=** and **SVC0=, SVC4=, SVC5=** operands no longer require values of **NONE** for TYPE=MCH|XTP REMOTEs in order to eliminate NAS1301W configuration warning messages. The values can now be coded as NONE or left blank.

12) **Configuration and Alert Message Reassignment** - The following HNAS messages were reassigned to improve category classification:

```
From      To       Message ID
--------  -------- --------------
NAS0010I->         'INITIALIZATION STARTED AT hh:mm:ss ON mm/dd/yy'
        ->NAS0000I 'HOST NAS STARTED AT hh:mm:ss ON mm/dd/yy'
```

13) **NAS0001I HOST NAS INITIALIZATION COMPLETE, ALL FUNCTIONS READY** message is now generated once the initialization phase of the HNAS activation process completes successfully. This new alert message is independent from the HNAS and router connectivity alert messages, like **NAS2020I** which was previously employed by users in an attempt to determine HNAS availability after the activation process. The new message now provides a more accurate representation of HNAS system availability.

14) **NAS1***s Configuration Definition File (CDF) scan /decode messages** are no longer written to the operator console when HNAS is started. The messages will continue to be

logged into the SYSPRINT file should viewing be required. In the unlikely event that you would like to continue receiving these message at the operator console you can re-enable using the PARM=**SHOWCNFG** option. **SHOWCNFG OFF** is the default start parameter and is new to V2R2M0.

15) **SHOWERR** is now the default start parameter which inhibits information only HNAS Alert messages from being displayed on the system console. Only HNAS error messages will be displayed although all messages will continue to be written to the SYSPRINT file. In Previous HNAS levels, SHOW ON (SHOWALL) was the default value which caused all Alert messages to be routed to the system console. Users who prefer to see select error and information messages should run with start parameter SHOWALL and filter unwanted information or error messages using the ALRMFLTR= option.

    Note: Starting with 220, console command output is, by default, **not routed** to SYSCONS. To override the default and allow console command output to be routed to SYSCONS, specify the SHOWON start parameter or the SHOW ON console command.

16) The **TRCPRNT** start parameter and **TRCPRNT ON** console command are **now also used to control the display/logging of NAS2nnnI** TCP/IP Event Alert Informational **messages**. These messages will no longer be displayed or logged in **TRCPRNT OFF** mode (TRCPRNT omitted from the HNAS Start Parameter or set off via TRCPRNT OFF console command).

    These informational messages are considered 'trace like' and are rarely required. Further more, they generate unnecessary Operator Console (when SHOWOFF or SHOWERR are omitted) and SYSPRINT activity. If TRCPRNT mode is required to capture other Alert Message activity we suggest that you eliminate these information messages using the ALRMFLTR=(SUPPRESS,NAS2***I(S),...) or ALRMFLTR=(PURGE,NAS2***I(S),...) alarm filter options. Note that TRCPRNT can be active when all other tracing is off to allow informational alert messages.

17) **TRCLU MINDATA, TRCVC MINDATA, TRCMCH** and **TRCMCHX** are now the default trace options enabled at HNAS activation. The trace entries will be written to the local trace table (see BUILD **TRCLMT=** operand) unless **TRCPRNT** option is enabled where trace records will be written to the SYSPRINT file.

    WARNING: The **TRCPRNT** option **consumes additional computing cycles** and can **influence HNAS performance and available Host CPU cycles** especially when several traces are active. We suggest that the option only be enabled under the guidance of a HNAS support representative. To eliminate all trace activity you can code **TRCALL OFF** on the HNAS start parameter or you can turn off specific tracing by specifying; **TRCLU OFF**, **TRCVC OFF**, **TRCMCH OFF** or **TRCMCHX OFF**.

18) **NAS6715W and NAS7715W Alert Message** SYSPRINT entries now include a display of the buffer content for the call request packet that encountered the clear condition.

19) **DNAS console output** HOST= was changed from OS390 to OS390|ZOS to identify as a common distribution for Z/OS and OS/390 environments.

20) The **DPARM** console display output was modified to list all of the default start parameter options to improve presentation of the operational parameters.

21) Execution of console commands specified on the **CONCMDQ=** BUILD operand will now be deferred until after the initialization process completes. This reduces intermingling of activation messages and console display command output.

22) **HNASBook** documentation manual was **reorganized** and **grouped** into **separate manuals** with some of the sections renamed to improve usability. Please refer to the following table on the next page for section cross referencing:

The following table denotes the documentation manual section reassignment:

HNAS 220 Manual	HNAS 220 Section	HNAS 211 Section	HNAS 220 Section Description
HNASBOOK	-	-	HNAS Configuration Guide and Reference.
-	Preface	Preface	Preface, Important Notes and New Features
-	Chapter 1	Chapter 1	Introduction
-	Chapter 2	Chapter 2	Installation and Activation
-	Chapter 3	Chapter 3	Configuration Statement Guide
-	Chapter 4	Chapter 4	Configuration Statement Ref.
-	Chapter 5	Chapter 7	Migration Overview
-	Chapter 6	Chapter 8	Maintenance (APAR and PTF)
-	Appendix A	Appendix F	X.3 Pad Parameter Summary
-	Appendix B	Appendix G	HNAS Configuration Examples
-	Appendix C	Appendix H	Router Checklist Overview *
-	Glossary	Glossary	Glossary
MSGCODES	-	-	HNAS Messages and Codes * Debugging Guide
-	ConfigMsgs	Appendix B	Configuration Messages
-	AlertMsgs	Appendix C	Alert Messages
-	ResetCodes	Appendix D	Reset Cause/Diagnostic Codes
-	ClearCodes	Appendix E	Clear Cause/Diagnostic Codes
-	CiscoMsgs	(new)	Cisco Messages related to HNAS
CONSOLE	-	-	HNAS Console Subsystem Operations Guide * 2
-	Console	Chapter 5	Console Subsystem

HNAS 220 Manual	HNAS 220 Section	HNAS 211 Section	HNAS 220 Section Description
CONSTRC	-	-	HNAS Console Subsystem Operations Guide with Trace Entry Formats *
-	Console	Chapter 5	Console Subsystem
-	Trace	Appendix A	Trace Entry Formats
INDEXALL	-	-	HNAS Master Index (Index of sections from all books)

* - Tools for Problem Diagnosis
2 - Console Subsystem Operations Guide is also provided without trace entries for those who plan to print the guide but don't require use of the 70 pages of Trace Entry Formats.

23) **NAS2nnns and NAS2nnnI TCPIP Alert Message Enhancement** - This new TCPIP alert message format may prevent host system automation traps or filters from ignoring or highlighting some HNAS TCPIP alert message events. System automation filtering options may require tweaking to accommodate the new alert message formats.

Please refer to the HNAS New Features - V2R2M0 entry for a description of the new message formats. <**06-25-2003** - Although this new feature was provided in the initial implementation of 220 we erroneously omitted the description from this section>.

24) **TRCVC MINDATA** and **TRCLU MINDATA** independent parameters implemented as well as various HostNAS console trace command corrections/refinements. Please refer to APAR <Ref: 2200047 08-04-2003> for additional information.

25) Configuration process for **LLC0** or **LLC5 SYSL=(,NULL=0** no longer defaults to **NULL/0** (as of APAR 2200057) and will now generate an error message during HNAS CDF scan preventing activation. Simply change **NULL=*n*** to **NULL/*n*** in the CDF to prevent the error condition. Please refer to APAR <Ref: 2200057 10-23-2003> for additional information.

26) Prior to APAR 2200058: **TRCALL STOP** and **TRCLU|VC STOP|ALLOFF** commands erroneously reset the global **TRCLU|VC MINDATA|MAXDATA** buffer data logging option forcing **NODATA** instead. **TRCALL STRT** and **TRCLU|VC ALLON** commands erroneously force the global **TRCLU|VC MAXDATA** buffer data logging option. Please refer to APAR <Ref: 2200058 10-27-2003> for additional information.

27) The **FASTRUN** process will now propagate VTAM operands that are specified on a TYPE=MXT REMOTE definition statement if that MXT is associated with a SLU in the SVC0= and/or SVC5= operand list. If no MXT is associated with an SLU entry, the VTAM operands are taken from the root TYPE=MCH|XTP REMOTE definition statement as in previous releases. In this way, the generated APPL statements for each SLU can have different VTAM operands (for example, MODETAB, DLOGMOD, etc.).

28) Changes to **HNAS clear diagnostic** byte for **UNBIND**: A normal UNBIND from the PLU will result in a clear diagnostic byte of 0 (was 140) while a non-normal UNBIND from the

PLU continues to get a diagnostic byte of 140. Please refer to APAR <Ref: 2200062 11-24-2003> for additional information.

Changes made to this distribution after the initial release date may also be found in the **220 Maintenance Summary** section of the **HostNAS V2R2M0 MAINTENANCE (APAR and PTF) INFORMATION** section on the Maintenance Web site.

Migration - V2R1M1

Program Logic changes to V2R1M1 that may affect your environment migration

1) **TAP=0** is the new default operand for TYPE=XOT|XTP REMOTE definitions. In previous releases the default TAP (keep-alive) value was 60. Some XOT router IOS or software levels don't always respond to an XOT Clear Request received on an idle virtual circuit but must respond to a XOT Call Request. <Ref: 2100015 05-16-2002>

2) **Alert Message Reassignment** - Duplicate alert message identifiers can cause confusion when analyzing event alert message activity. <Ref: 2100020 05-31-2002> The following alert messages were reassigned to eliminate duplicate assignment and improve group category classification:

```
  From       To      Message ID
--------  -------- --------------
NAS2321W->NAS2501W (KEEPALIVE FAILED)     (was dup)
NAS2322E->NAS2502E (CONTACT LOST)
NAS2323I->NAS2503I (CONTACT REAQUIRED)
NAS2321W->NAS2401W (RECEIVE FAILED)       (was dup)
NAS2331W->NAS2411W (SEND FAILED)
NAS2311W->NAS2331W (IOCTL FAILED)
NAS2281W->NAS2291W (SETSOCKOPT FAILED)
NAS2261W->NAS2281W (GETSOCKNAME FAILED)
NAS2260I->NAS2280I (GETSOCKNAME COMPLETE) (was dup)
```

3) **DNAS console output** no longer displays the DTR= (VOLSER) value. This information is no longer on any use. <Ref: 211_internal 08-26-2002>

Migration - V2R1M0

Program Logic changes to V2R1M0 that may affect your environment migration

1) HNAS Alert Message NAS0101W (Buffer Shortage) updated. This message now shows the current free buffer count as well as the maximum free buffer count. It is issued when the buffer pool becomes 60% depleted. If you continually receive this alert message, consider increasing the number of buffers in the HNAS buffer pool (see BFRLMT operand on BUILD definition statement).

2) Custom modifications to replace the ***calling*** DTE address in a GATE **outbound** Call Request packet (GATE=GENERAL/REPDCEADDR) or to strip the ***called*** DTE address from a GATE **inbound** Call Request packet (GATE=GENERAL/STRIPRTEIN) have been incorporated into the standard logic. To request these features, specify:

   For V1R1M4: GATE=GENERAL/REPDCEADDR
   For V2R1M0: GATE=GENERAL,OPTIONS=REPDCEADDR
   For V1R1M4: GATE=GENERAL/STRIPRTEIN
   For V2R1M0: GATE=GENERAL,OPTIONS=STRIPRTEIN
   For V1R1M4: GATE=GENERAL/REPDCEADDR/STRIPRTEIN
   For V2R1M0: GATE=GENERAL,OPTIONS=(REPDCEADDR,STRIPRTEIN)

3) Custom modifications to strip the facilities field from a GATE **inbound** Call Request packet has been incorporated into the standard logic. To request this feature, specify:

   For V1R1M4: not supported
   For V2R1M0: GATE=GENERAL,OPTIONS=STRIPFAC

4) MBITCHN=NO parameter default was changed to MBITCHN=YES to mirror the IBM NPSI default chaining values.

5) XOT Call-out logic - Call Accept packet facilities from router are now processed by HNAS. This support allows the router to step down the facilities values (like packet and window sizes originally provided in the HNAS Call Request packet) allowing an additional attempt at facilities negotiation.

6) XOT LLC0/LLC5 Call-in logic - HNAS Call Accept packets now provide Calling/Called length and addresses in the outbound Call Accept packet.

   This global enhancement was disabled on April 28, 2002 (see APAR 2100002 for details) because some TRANSPAC X.25 implementations (and possibly other older X.25 network and PAD subscription implementations) don’t support DTE addresses in the X.25 Call Accept packet. This support will be provided via parameter OPTIONS= ECHODTEADDR in our upcoming V2R2 release. We apologize for any confusion regarding this feature.

7) The TAP=15 parameter default was changed to 0 (none) on May 16, 2002 (see APAR 2100015). This new default will accommodate router environments that do not properly support our tapping (keep alive simulation) option and allow those that do to define their preferred tapping value.

Migration - V1R1M4

Program Logic changes to V1R1M4 that may affect your environment migration

1) As of release V1R1M4, the console input prompt no longer includes the text from the NASNAME operand on the BUILD definition statement unless you request this via the PFXWTO start parameter or PFXWTO ON console command. This was done because the console input prompt is now configured using the CONPRMT operand on the BUILD definition statement.

2) As of release V1R1M4, the inactivity time-out value is now configured using the IDLETO operand on the BUILD definition statement. In prior releases, this value was fixed internally at 15 minutes. You may now vary this value between 0 (inhibit time-out) and 9999 minutes. (09-20-2000) If omitted, a value of 0 will be used. (03-28-2001)

3) HNAS SYSPRINT BLKSIZE was optimized for MVS (08-09-2001).

4) HNAS Alert Messages; NAS3799I (Clear codes added) and NAS3702W (REQSESS Failed) modified/added (10-20-2001).

5) HNAS Alert Messages; NAS6715W and NAS7715W modified, NAS6716W, NAS6717W, NAS7716W and NAS7717W added (12-06-2001).

6) HNAS Alert Messages; NAS4700W, NAS4701W and NAS4702W added (01-16-2002).

CHAPTER 6

Maintenance (APAR/PFT)

This page left intentionally blank.

Chapter 6 - Maintenance Information

This chapter describes general maintenance information for the HNAS product. As with any software product, periodic maintenance is required to insure that users are protected from potential logic, timing or system level dependencies encountered after the initial product was released.

While the majority of HNAS documentation is downward compatible with earlier releases, the maintenance description/installation in this Chapter is not compatible with HNAS 230 or earlier. Please refer to your specific 230 product level guide for additional information.

Information Sources

Product Maintenance and Support Information and Documentation links are available on our Web site and FTP server. The HNAS PDF documentation manuals, maintenance (PTFs) and Refresh/Upgrade product eDistributions are available on our FTP server (userid required).

WEB Site Product Maintenance and Support Links

Current Product Maintenance and Support links are located at the following web site address:

http://www.comm-pro.com/hostnas/maint/index.htm

The maintenance web site contains the following HNAS Product Maintenance and Support information links:

<u>Product Notices</u>
Notices regarding HNAS product or component support issues

<u>Product Documentation Information</u>
Provides customers with up-to-date information regarding HNAS documentation change control activity for new or updated documentation manuals and sections.

<u>Product Features List by Reference</u>
The Product Features List by Release area provides a basic list of the key features and enhancements that were added to the respective HNAS releases. The Product's General Availability Date is also included for each of the denoted releases.

<u>Problem Summary Information</u>
Reference list for problems (currently being analyzed, pending an action or pre APAR assignment) or closed problems (non APAR solution provided or resolution deferral to a later HNAS release). The problem summary entries have the same structure as APAR memo's for potential implementation as standard APAR memos.

APAR Assignment Cross Reference List
Provides a matrix index denoting APAR numbering across HNAS distribution levels.

Maintenance Summaries (APAR/PTF)
Reference list of APARs and General Enhancements by release level. These files can be searched for information on ABENDs or other problem reports. Normally, maintenance fixes/updates are provided in the form of macro and/or object module replacements which are then added to the HNASMACX and/or HNASOBJX libraries, respectively. In rare cases, a ZAP might be provided in emergency situations when time is of the essence.

WEB Site Product Documentation Links

Current Product Documentation links are located at the following web site address:

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

This Web based Product Documentation Information area provides customers with up-to-date information regarding HNAS documentation change control activity for new or updated documentation manuals and sections.

Documentation for HNAS product updates or new releases is available in PDF book format (complete manual or section) or in printed format. Items with hyperlinks (highlighted and underlined) are available directly from the respective web page link. All other documentation is available on our FTP server (userid required) or can be obtained by contacting a HNAS marketing and support representative for an alternate form of documentation delivery. FTP is the preferred delivery method for our product documentation although E-mail (PDF file attachment) or mail/courier for physical media are also available.

FTP Server Links

Product Maintenance and Support links/file are located at the following FTP Server address:

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

The following HNAS documentation and maintenance directories are listed under all FTP userid root directories:

\hnas_doc <- **\hnas*vrm*d** sub-directories contain VRM specific PDF documents.
\hnas_maint <- **\hnas*vrm*m** sub-directories contain VRM specific PTF maintenance.

The following ‘user specific’ directories are provided under each FTP userid root directory for user and support file exchange:

\fromsup <- location of eDistribution directories or files from support (to user).
\tosup <- location of directories or files to support (from user).

FTP Server access requires a HNAS userid which can be obtained by contacting your HNAS marketing and support representative.

Maintenance Overview

Up-to-date HNAS product maintenance information is available on our Product Maintenance and Support Web Site as well as our FTP Server (see Information Sources section of this document for specific location details). The online Maintenance and Support documentation information page contains content and links to a variety of information such as: Product Notices, Problem Summaries, APAR Summaries, V*n*R*n*M*n* Maintenance Summaries, Custom Enhancements (made available after the initial product release), as well as other problem description, resolution, support and maintenance information.

Product maintenance APAR information and circumventions for specific APARs can be accessed on our Support Web Site or FTP Server.

Sometimes HNAS APARs or general maintenance requires that the product be reinstalled with a refresh or upgrade containing all current maintenance. This is often the simplest approach to installing cumulative maintenance.

Product upgrade and refresh distributions are provided as requested by users through their HNAS product support channels. Support channels must be used to order SMP/E maintenance because the MCS (Modification Control Statements), which are unique to each customer, are prepared by Comm-Pro.

Product Maintenance Identification

HNAS product distribution and maintenance level can be identified based on the generation library names as well as querying the HNAS operator console. The HNAS DNAS console command provides product distribution level information while the '**DNAS APAR**' and '**DMAP APAR**' commands provide specific APAR maintenance level information.

Please refer to the **DNAS** and **DMAP** display commands located in the Console Command section of the HNAS Console Subsystem Operations Guide for additional information.

Custom Maintenance Identification

HNAS custom product distribution and maintenance information can be identified based on CustomMod, Custom-User-Mod or Problem-ID assignment identification (internally referred to as XFFIX ID's). The HNAS '**DNAS USERMOD**' console command display provides a list of these identification types on your system. Custom enhancements are provided to users in product refresh or upgrade distributions or as special PTF type packaging via ftp or e-mail.

The majority of customer requested HNAS product enhancements are now typically implemented as standard Enhancement APARs so custom maintenance is less frequent.

Please refer to the **DNAS** display command located in the Console Command section of the HNAS Console Subsystem Operations Guide for additional information.

Product Maintenance by Distribution Types

Standard (non-SMP/E) Distribution

TSO Transmit format is our standard method of delivering the data sets required for HNAS maintenance. Most of our customers prefer this format because of ease of product installation. .str files created by TSO TRANSMIT commands are placed in a zip file which is distributed to the customer. The zip file is unzipped (usually on a PC) and the .str files are sent to the mainframe (FTP or IND$FILE) where TSO RECEIVE commands are used to create data sets with corrected HNAS components.

SMP/E Distribution

Product maintenance is usually in the form of cumulative PTFs that replace all HNAS components. The SYSMOD-ID of a replacement PTF will have the form Svrmaaa where vrm is the HNAS version, mod, release level and aaa is the highest numbered APAR in the PTF (e.g. S250011). Comm-Pro supplies MCS (Modification Control Statements) required to install a replacement PTF. In special cases individual APARs may be packaged for SMP/E installation using the SYSMOD-ID of the form Tvrmaaa. APARs are not usually provided because of the difficulty of preparing MCS for different APAR sets at every SMP/E installation.

Product Maintenance Types

Upgrade Product

This method of maintenance application is employed when a user upgrade to a more current HNAS V*n*R*n*M*n* release (for example from V2R3M0 to V2R4M0). This is generally the best approach in keeping maintenance current and provides new or expanded features not available in older HNAS product releases.

Refresh Product

This (preferred over Object and Source PTF’s) method of maintenance is employed when users prefer to bypass installation of individual APARS or are required to include APARs that are only available on a refresh distribution. With HNAS product refreshes, the most recent (up-to-date) maintenance distribution is based on the users existing installed HNAS V*n*R*n*M*n* release level. This is the best approach in keeping maintenance current when there is no interest or requirement to upgrade to a newer release. See the Upgrade and Refresh Distributions section later in this chapter for additional information.

Object Module and Source Replacement Members

This method of maintenance distribution is provided for customers who prefer to apply a specific APAR to address a maintenance concern without having to apply a product refresh. The object or source replacement members are provided on the FTP Server in their respective HNAS V*n*R*n*M*n* APAR maintenance sections. If required, APARs for SMP/E installations can be custom built by Comm-Pro (because MCS is required).

ZAP (Temporary Problem Fixes)

This method of maintenance application is employed only in emergency situations when users require an immediate fix for a specific problem. While Object and Source replacement members are generally the preferred method of maintenance installation for individual APAR PTF application, in some cases users may find the installation of a ZAP the simplest problem fix. If a ZAP is available, IMASPZAP or AMASPZAP JCL and control cards will be provided by Comm-Pro on an as needed basis. ZAPs for SMP/E installations are custom built by Comm-Pro (because MCS is required).

End of Trial Period Authorization Date (EOTDATE) Extension via EOTKEY

The HNAS product End of Trial Authorization Date (EOTDATE) specifies when the HNAS Trial Period 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 refresh distribution. The new **EOTKEY=*dd...dd*** start parameter now allows the EOTDATE to be extended without the need of a new refresh distribution. The ***dd...dd*** digits represent an encoded 16 decimal digit key. The EOTKEY=*dd*...*dd* digits are provided by Comm-Pro in a special file that is sent as an E-mail attachment or is downloaded from our FTP server. The EOTKEY=*dd*...*dd* string can then be cut and pasted to the PARM= operand (or MMEM console command). 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
```

The DNAS display for a trial distribution has the following form (for example):

```
HOST NAS INFORMATION FOLLOWS
     HNAS VERSION=V2R4M0 DIST=SMP/E                            1
     HNAS PROGRAM RUNNING UNDER z/OS 01.11.00                  2
     HNAS PRODUCT INSTALLED UNDER z/OS 01.11.00                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 2010/11/29                   8
HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400106       9
MOST RECENT MAINTENANCE APPLIED IS APAR 2400106                 10
AUTH=032D SHIPID=1100000011199999                               11
CUSTID=SFD_99999                                                12
CUSTINFO=COMM-PRO ASSOCIATES                                    13
TRIAL PERIOD EXPIRATION DATE IS 2010/12/31                      14
DATAFONO SUPPORT IS INCLUDED                                    15
                                                                16
                                                                17
APARID   MAINTENANCE STATUS                                     18
ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400106             19
```

If EOTKEY=4961000737880526 (for example) is used to extend the EOTDATE, DNAS display lines 11, 14 and 16 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                            16
```

Note: The EOTKEY= (End of Trial Authorization Date Extension Key) method of updating the EOTDATE was introduced into 240 by APAR 2400106. Prior to APAR 2400106, an HNAS Upgrade or Refresh was required to extend the EOTDATE.

End of Maintenance/Use Anniversary Date (EOMDATE) Extension via EOMKEY

The HNAS product Maintenance/Use Anniversary Date (EOMDATE) specifies when the HNAS 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 **EOMKEY=*dd...dd*** start parameter now allows the EOMDATE to be extended without the need of a new refresh distribution. The ***dd...dd*** digits represent an encoded 16 decimal digit key. The EOMKEY=*dd...dd* digits are provided by Comm-Pro in a special file that is sent as an E-mail attachment or is downloaded from our FTP server. The EOMKEY=*dd...dd* string can then be cut and pasted to the PARM= operand (or MMEM console command).

Note: The EOMKEY= (End of Maintenance/Use Anniversary Date Extension Key) method of updating the EOMDATE was introduced into 240 by APAR 2400095 and then modified by APAR 2400096. Prior to APAR 2400095, an HNAS Upgrade or Refresh was required to extend the EOMDATE.

Effective with APAR 2400095, the EOMKEY file has the following format:

```
EOMKEY=1760921617519627
HNAS EOMKEY CREATED AT 16:51:10 ON 2009/08/13
SHIPID=0100000000199999
CUSTID=SFD_99999
CUSTINFO=COMM-PRO ASSOCIATES
```

```
MAINTENANCE/USE ANNIVERSARY DATE IS 2009/08/31
```

Effective with APAR 2400096, the EOMKEY file has the following format:

```
EOMKEY=1760931717683627
HNAS EOMKEY CREATED AT 18:13:36 ON 2009/09/04
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/02/01
CUSTID=SFD_99999
CUSTINFO=COMM-PRO ASSOCIATES
EMKYDC=7201002011199999
```

The DNAS command output (display line 12) depicts the Maintenance/Use Anniversary Date (EOMDATE) that was in effect when the eDistribution was generated (initial, upgrade or refresh) or when the last EOMDATE Extension Key (EOMKEY) was applied:

MAINTENANCE/USE ANNIVERSARY DATE IS *yyyy*/*mm*/*dd*

When EOMKEY=*dd*...*dd* is used to extend the EOMDATE, an asterisk (*) will follow the date. For example:

MAINTENANCE/USE ANNIVERSARY DATE IS 2010/08/31*

It's important that Prepaid Maintenance and Purchase Order periods are kept up-to-date to ensure that a current EOMDATE is provided in a timely manner by your HNAS support organization to avoid NAS9206*i* alert message generation or a lapse in Maintenance/Use service.

Trial Distribution to Permanent Distribution Conversion via EOMKEY

The **EOMKEY=*dd*...*dd*** start parameter or **MMEM EOMKEY=*dd*...*dd*** console command also allows a trial distribution to be converted 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 16 above as follows:

```
AUTH=000  SHIPID=1100000011199999 EMKYID=1100000011199999      11
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31*                14
EOMKEY=4962030747980516 IS IN EFFECT                           16
```

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.

Product Maintenance Considerations

Every effort has been made to preserve previously defined default parameter and option values as well as the operation characteristics of the code so that installation of APARs whether individual APAR or via Refresh distribution occurs with minimal, if any impact to our customers. There are times, however, when the program logic must be updated or altered for new program options or to correct previously defined default assumptions found to be incorrect. This includes changes to configuration operands, messages and runtime processes that can cause the environment to operate differently than previous APAR maintenance levels.

We highly recommend that you review the migration section for the release that you are applying maintenance to. This information can be found in Chapter 5 (Migration) denoted under the respective V*n*R*n*M*n* release heading **“Program Logic changes to V*n*R*n*M*n* that may affect your environment migration”** of this documentation manual.

It is also highly recommended that once the maintenance is installed to your respective HNAS V*n*R*n*M*n* level that the first activation test be done in FASTRUN mode.

Product Maintenance Installation (non-SMP/E)

Upgrade and **Refresh** maintenance is installed using the procedures in Chapter 2 to create a new set of HNAS libraries and to generate a new HNAS load module. We strongly advise that users install the upgraded product into HNAS @*vrmnnnn* identified libraries.

For **APAR** maintenance, the **HNASMNT** job is used to rebuild the HNAS load module after updated object or source members are placed in the hlq.HNASMACX and hlq.HNASOBJX product maintenance libraries. The **HNASMNT** job is one of the jobs built in the hlq.HNASCNTL PDS by the HNASGJOB REXX EXEC.

For all maintenance processes, please ensure that appropriate HNAS library distribution generation naming conventions and/or library back-up policies are implemented to preserve the integrity of the production HNAS libraries until all testing is completed and stable production implementation is achieved.

Upgrade and Refresh Distribution Maintenance Installation (non-SMP/E)

Distribution files are provided as a single compressed zip archive file which contains the required libraries. The HNAS product refresh or upgrade distribution files are located on our FTP server userid accounts (FTP access requires a customer userid) or via special request e-mail file attachment for registered HNAS users.

HNAS customers should contact their customer support representative via telephone or e-mail to request that a refresh or upgrade distribution be prepared for their environment. HNAS authorization logic requires that a unique SHIPID_KEY distribution be created for each HNAS customer.

Standard installation instructions should be followed when installing HNAS Product upgrades or refreshes. We advise that you review the migration information in Chapter 5 and the new features information in Appendix D of the HNAS Guide and Reference documentation to identify potential CDF configuration requirements for new parameters or changes in existing parameters.

The HNASGJOB REXX EXEC provided with each refresh or upgrade distribution must be used to ensure that the installation process is up-to-date. This prevents maintenance conflicts when back level source and object libraries are sometimes erroneously referenced in old JCL or libraries.

Newly installed distributions should be installed in a unique set of distribution libraries preserving the production or earlier generation level libraries until testing and production implementation is completed. We suggest that you create a second level DSN qualifier of @*vrmnnnn* (Version, Release, Modification and maintenance level id) as part of the library name.

Upgrade and refresh distributions are contained in a zip file named as follows:

hnas_*vrmnnnn_yyyy-mm-dd*_cust#_custid.zip

Example: hnas_2400011_2006-04-04_90000_cpt.zip)

This file contains a complete HNAS system at the indicated base (vrm) and APAR (nnnn) levels. The HNAS files contained in the zip file are installed using the procedures in Chapter 2. If the *hlq* used for HNAS data sets is SYSX then we recommend that the dataset names created by the HNASGJOB EXEC have the form *hlq*.@vrmnnnn.HNAS*type* so that the distribution level is obvious from the data set name. (HNAS*type* = HNASCNTL, HNASLOAD, HNASMAC, HNASMACX, etc.).

Important reminders:

If you are not using the HNASXEQ JOB built in hlq.HNASCNTL by the HNASGJOB REXX EXEC then your job or PROC must be updated to use the new libraries.

When upgrading from an older level be sure to copy the CDF file into the a new source file location in the event changes are required for the upgrade.

The new HNASXEQ JCL or PROC must point at the new configuration file location.

Be sure that the AUTH DD statement in the job that invokes HNAS points at the NASAUTH member in the new HNASMAC library. If the wrong library is used a NAS9203S alert message is issued and operations will terminate with:

HALT AT LOC xxxxxxxx IN NAS9203S: HNAS AUTHORIZATION FILE IS INVALID...,

Please refer to the product Installation section in Chapter 2 and product migration considerations in Chapter 5 of the HNAS Guide and Reference for additional information.

Object Module and Source Member Maintenance Installation (non SMP/E)

APAR Memo entries are provided as HTML Maintenance Summary files on our WEB Site. On our FTP Server, APAR Memos and Object (OBJ) and/or Source (SRC) replacement members are provided in a single *vrmnnnn*.zip (e.g. 2400077.zip) archive file. The archive file contains ASCII text (*.txt), TSO XMIT Stream (*.str) and EBCDIC binary (*.bin) format files. The zip files can be unzipped on your PC or Host system running with the appropriate unzip software.

<u>OBJ/SRC FTP directory and file structures</u>

\hnas_maint <- maintenance root directory listed under all FTP userid accounts
\hnas*vrmm* <- sub-directories contain VRM specific PTF maintenance.
***vrmnnnn*.zip** <- ZIP APAR files are located under the HNAS V*n*R*n*M*n* level.
Unzipping the above produces the following:
***vrmnnnn*_HNASMEMO.TXT** <- APAR PTF memo file in ASCII format.
***vrmnnnn*_HNASMEMO.BIN** <- APAR PTF memo file in EBCDIC format.
***vrmnnnn*_HNASOBJX.STR** <- APAR OBJ stream file in TSO XMIT format. *

***\vrmnnnn*_HNASMACX.STR** <- APAR SRC stream file in TSO XMIT format. *

*** -** APAR distributions may contain OBJ and/or SRC maintenance. Please refer to the APAR Memo PTF_TYPE: field for PTF type requirements.

Based upon the structure above, the APAR for 2400077 would be located as follows:

\hnas_maint\hnas240m\2400077.zip

Object and Source Maintenance Installation Steps (non-SMP/E)

1) Back up the following data sets (preferred second level DSN qualifier format assumed):
hlq.@*vrmnnnn.*HNASLOAD
hlq.@*vrmnnnn.*HNASOBJX
hlq.@*vrmnnnn.*HNASMACX

HNASLOAD contains the HNAS load module currently in use.
The HNASOBJX and HNASMACX data sets receive revised object and source members. If there are problems with the new members then the old system can be recovered by restoring the above data sets.

2) Download hnas_maint/hnas*vrm*m/apars/*vrmnnnn*.zip from our FTP server to a file named vrmnnnn.zip on a staging PC.

3) UNZIP this file to reveal the APAR memo files in ASCII and EBCDIC (.TXT and .BIN) and the HNASOBJX.STR and HNASMACX.STR files. HNASMACX.STR will be present only if the APAR has a SRC component.

4) Upload *vrmnnnn*_HNASOBJX.STR and *vrmnnnn*_HNASMACX.STR to your host using a BINARY file transfer. The files should be placed in hlq.@*vrmnnnn*.HNASOBJX.STR and hlq.@*vrmnnnn*.HNASMACX.STR. Any names will do (these work files are only used in Steps 5 and 6 below).

Dataset attributes: BLKSIZE=3200, LRECL=80, DSORG=PS,RECFM=FB, SPACE=(TRK,(10,10)) will be sufficient.

5) If hlq.*vrmnnnn*.HNASOBJX.STR was uploaded issue the TSO command:
RECEIVE INDS('hlq.@*vrmnnnn*.HNASOBJX.STR').

When restore parameters are requested enter:
DSN('hlq.@*vmrnnnn*.HNASOBJX')

This will load the OBJ PDS members for the APAR.

6) If hlq.*vrmnnnn*.HNASMACX.STR was uploaded issue the TSO command:
RECEIVE INDS('hlq.@*vrmnnnn*.HNASMACX.STR').

When restore parameters are requested enter:
DSN('hlq.@*vmrnnnn*.HNASMACX')

This will load the SRC PDS members for the APAR.

7) Run the HNASMNT JOB in hlq.@*vrmnnnn*.HNASCNTL. Since hlq.@*vrmnnnn*.HNASOBJX and hlq.*vrmnnnn*.HNASMACX are concatenated in front of the standard distribution libraries (HNASOBJ and HNASMAC) in the HNASMNT job, the members installed in steps 5 and 6 are used to build the new HNAS load module.

Sample HNASMNT Maintenance JCL File

The following JCL is a sample of the HNASMNT non-SMP/E maintenance JOB. This JOB is built as member HNASMNT in hlq.HNASCNTL by the HNASGJOB REXX EXEC. The hlq (SYSCPT.@2400000) is supplied when the HNASGJOB EXEC is customized to reflect installation requirements.

```
//HNASMNT  JOB ACNT#,'(RE)BUILD HNAS',CLASS=A,
// MSGCLASS=A,NOTIFY=&SYSUID,REGION=0K
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       JOB TO BUILD (OR REBUILD) HNAS LOAD MODULE
//*
//*       THIS JOB IS RUN AFTER THE INITIAL INSTALLATION OF HNAS FILES
//*       AND AFTER MAINTENANCE HAS BEEN PLACED INTO THE HNASMACX
//*       AND/OR HNASOBJX DATA SETS.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       STEP 1: ASSEMBLE MODULES WITH TCP/IP OR VTAM DEPENDENCIES
//*               USING INSTALLATION'S TCP/IP & VTAM LIBRARIES.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//        JCLLIB ORDER=SYSCPT.@2400000.HNASCNTL
//NASMAIN EXEC NASASM,OBJ=NASMAIN
//ASM.SYSIN DD *
         NASMAIN HOST=ZOS
         END
/*
//NASTCP  EXEC NASASM,OBJ=NASTCP
//ASM.SYSIN DD *
         NASTCP HOST=ZOS
         END
/*
//VTMEXIT EXEC NASASM,OBJ=VTMEXIT
//ASM.SYSIN DD *
         COPY  VTMEXIT
/*
//VTMRCV1 EXEC NASASM,OBJ=VTMRCV1
//ASM.SYSIN DD *
         COPY  VTMRCV1
/*
//VTMSND1 EXEC NASASM,OBJ=VTMSND1
//ASM.SYSIN DD *
         COPY  VTMSND1
/*
//VTMSND2 EXEC NASASM,OBJ=VTMSND2
//ASM.SYSIN DD *
         COPY  VTMSND2
/*
//VTMTR   EXEC NASASM,OBJ=VTMTR
//ASM.SYSIN DD *
         COPY  VTMTR
```

```
/*
//VTMUT1  EXEC NASASM,OBJ=VTMUT1
//ASM.SYSIN DD *
         COPY  VTMUT1
/*
//*
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*       STEP 2:  CREATE THE HOST NAS LOAD MODULE.
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//BUILD  EXEC  PGM=IEWL,PARM='LIST,LET,SIZE=(384K,96K),AC=1'
//SYSPRINT DD  SYSOUT=*
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(256,20))
//SYSLIB   DD  DSN=TCPIP.SEZACMTX,DISP=SHR
//SYSLMOD  DD  DSN=SYSCPT.@2400000.HNASLOAD,DISP=OLD
//HNASOBJ  DD  DSN=SYSCPT.@2400000.HNASOBJX,DISP=SHR
//         DD  DSN=SYSCPT.@2400000.HNASOBJ,DISP=SHR
//SYSLIN   DD  DSN=SYSCPT.@2400000.HNASMACX(SYSLIN),DISP=SHR
/*
```

Product Maintenance Installation (SMP/E)

Maintenance distribution files are provided as a single compressed zip archive file which contains the required files. The distribution files are located on our FTP server userid accounts (FTP access requires a customer userid) or via special request e-mail file attachment for registered HNAS users.

HNAS customers should contact their customer support representative via telephone or e-mail to request that a refresh or upgrade distribution be prepared for their environment.

The SMP/E MCS (Modification Control Statements) required to install maintenance is provided by Comm-Pro.

Prior to installing an upgrade or refresh distribution we advise that you review the migration information in Chapter 5 and the new features information in Appendix D of the HNAS Guide and Reference documentation to identify potential CDF configuration requirements for new parameters or changes in existing parameters.

Upgrade uses the procedures in Chapter 2 to install a new HNAS release (vrm) with a new FMID (LNS0vrm).

The distribution filename has the following format:
lns_vrmnnnn_yyyy-mm-dd_cust#_cid.zip

Where:
lns indicates that this is an SMP/E distribution.
vrm = HNAS version, mod, release level.
nnnn = Highest number APAR in the upgrade.
cust# = Comm-Pro assigned 5 digit customer number
cid = Comm-Pro assigned 3 character customer ID string.

Example: lns_2400011_2006_03_13_07003_cpt

Refresh in a New SMP Base Library (smp/e refresh reinstall) maintenance is used when a customer prefers to load a new distribution into a new SMP library set. This avoids the need to install a refresh PTF. The procedure for loading a ‘new’ system is described in Chapter 2 and is basically the same as described in **Upgrade** (above).

Refresh maintenance uses a collection of jobs to install a refresh PTF which replaces all HNAS components under their existing HNAS release (vrm) to ensure that all APARs are present. The jobs (RECEIVE, APPLY & ACCEPT) are built by the install process.

The distribution filename has the following format:
lns_Svrmaaa_yyyy-mm-dd_cust#_cid.zip
Where:
lns indicates that this is an SMP/E distribution.

Svrmaaa is the SYSMOD ID of the replacement PTF
  vrm   = HNAS version, mod, release level.
  aaa   = highest APAR number in the PTF.
             Note: APAR memos use a 4 digit number (nnnn).
  cust# = Comm-Pro assigned 5 digit customer number
  cid   = Comm-Pro assigned 3 character customer ID string.

Example: Ins_S240017_2006_03_13_07003_cpt

Object and **Source** maintenance installation uses a collection of jobs (RECEIVE, APPLY, ACCEPT, built by the install process) to install APARs.

The distribution filename has the following format:
Ins_Tvrmaaa_yyyy-mm-dd_cust#_cid.zip
Where:
Ins indicates that this is an SMP/E distribution.
Tvrmaaa is the SYSMOD ID of the individual APAR.
  vrm   = HNAS version, mod, release level.
  aaa   = APAR number.
             Note: APAR memos use a 4 digit number (nnnn).
  cust# = Comm-Pro assigned 5 digit customer number
  cid   = Comm-Pro assigned 3 character customer ID string.

Example: Ins_T240017_2006_03_13_07003_cpt

This type of maintenance is only used in under unusual circumstances. The reason for this is that Comm-Pro is not in a position to provide the MCS required to install individual APARs at each of our SMP/E installations.

Refresh / Object and Source Installation APAR Steps (SMP/E)

SMP/E refresh distributions for HNAS at the APAR aaa level are installed with an SYSMOD ID of Svrmaaa. The required files are in Ins_Svrmaaa_yyyy-mm-dd_cust#.cid.zip

An Object and Source distribution for HNAS APAR aaa is installed with a SYSMOD of Tvr-maaa. The required files are in Ins_Tvrmaaa_yyyy-mm-dd_cust#.cid.zip

To install refresh PTF S240aaa:

Unzip Ins_S240aaa.zip to create on a PC to create:
S240aaa_ptfmac.str
S240aaa_ptfsrc.str
S240aaa_ptfobj.str

1) On Z/OS allocate:
hlq.S240aaa.PTFMAC.STR

DSORG=PS,RECFM=FB,BLKSIZE=3120,LRECL=80,SPACE=(TRK,(100,10))
hlq.S240aaa.PTFSRC.STR
DSORG=PS,RECFM=FB,BLKSIZE=3120,LRECL=80,SPACE=(TRK,(30,10))
hlq.S240aaa.PTFOBJ.STR
DSORG=PS,RECFM=FB,BLKSIZE=3120,LRECL=80,SPACE=(TRK,(30,10))

2) Copy:
S240aaa_ptfmac.str to ***hlq***.S240aaa.PTFMAC.STR using BINARY mode (IND$FILE or other transfer program).
S240aaa_ptfsrc.str to ***hlq***.S240aaa.PTFSRC.STR using BINARY mode (IND$FILE or other transfer program).
S240aaa_ptfobj.str to ***hlq***.S240aaa.PTFOBJ.STR using BINARY mode (IND$FILE or other transfer program).

3) Use TSO RECEIVE INDS('....') to create and load the following PDSs:
hlq.S240aaa.PTFMAC from ***hlq***.S240aaa.PTFMAC.STR
hlq.S240aaa.PTFSRC from ***hlq***.S240aaa.PTFSRC.STR
hlq.S240aaa.PTFOBJ from ***hlq***.S240aaa.PTFOBJ.STR

The above PDSs can be pre allocated, if required, using sizes for ALNSMAC, ALNSSRC and ALNSOBJ. If DSN(...) UNIT(...) and VOLUME(...) are entered when RECEIVE prompts for restore parameters then RECEIVE will allocate correct sizes.

The *.STR files are no longer needed at this point.

hlq.S240aaa.PTFMAC(S240aaaM) has the MCS for the refresh PTF.

4) RECEIVE S240aaa
SMP.PTFIN:
//SMP.SMPPTFIN DD DSN=***hlq***.S240aaa.PTFMAC(S240aaaM),DISP=SHR

5) APPLY S240aaa
The following DD statements are required:
//S240aaaM DD DSN=***hlq***.S240aaa.PTFMAC,DISP=SHR
//S240aaaS DD DSN=***hlq***.S240aaa.PTFSRC,DISP=SHR
//S240aaaO DD DSN=***hlq***.S240aaa.PTFOBJ,DISP=SHROur FTP server has

After the SMP APPLY step PTFMAC and PTFSRC members are in the SMP target libraries (***hlq***.SLNSMAC and ***hlq***.SLNSSRC). ***hlq***.SLNSMAC (and PTFMAC) will contain the new NASAUTH file for the distribution. The AUTH DD statement in the job that invokes HNAS must point at the new NASAUTH member (or a copy of the member) to avoid a NAS9203 alert message for an invalid authorization file. The NASAUTH member in a refresh PTF distribution is unique to that distribution and cannot be used for other distributions.

The procedure to install APAR aaa is identical to the above except that S240aaa is replaced by T240aaa. APAR distributions will not normally contain a NASAUTH member).

Trial Distribution Extension (NASAUTH) Installation

Commencing with the V2R4M0 release of HNAS product we no longer provide product trial distribution extensions via a renewed NASAUTH file. Should a trial extension be required, a trial refresh distribution can be made available for installation (based upon the distributor or business partner trial agreement) or a permanent distribution can be purchase.

Trial product distribution refresh extensions are installed in the same manner as are standard product refresh (see related topics in the documentation Guide).

Staring with V2R4M0 APARs 2400095, 2400096 and 2400106, start parameters are now provided that allow a trial expiration date to be extended dynamically or a trial distriubution to be converted to a permanent distribution dynamically. For more information, please see the description of the EOTKEY and EOMKEY start parameters in Chapter 2 of this dociument.

APAR/PTF Memo Overview

Unique APAR’s (Authorized Program Analysis Report) records provide the maintenance level ***vrm*** and control numbers ***nnnn*** that are assigned as a reference to identify problems or anomalies that were encountered under supported HNAS distribution levels (i.e. **2400001**). Each APAR contains control dates and fields, problem description, solution or circumvention information. The PTF (Program Temporary Fix) may be identified as specific apply information for patches, source or object member replacement and in some cases may require that a maintenance refresh be applied. Some non-critical APAR’s may be deferred to the next release level which could be a Version, Release or Modification level distribution.

Note: APAR’s denoted with **STATUS: ‘CLOSED_DEFERRED to release VnRnM0’** indicate that no corrective logic (PTF) was provided for the respective APAR under the HNAS release maintenance level ***vrm*** for the control numbers ***nnnn*** that was assigned for the problem. Corrective logic will be provided in the denoted ‘**to release VnRnM0**‘ or a later release.

APAR Memo Format - New Layout

The following is a sample APAR memo maintenance control record:

```
yyyy-mm-dd  - APAR vrmnnnn     (memo date, APAR vrm and nnnn number)

       APAR:  vrmnnnn          (APAR # e.g. 2400001)
     STATUS:  OPEN|PENDING|TESTING|CLOSED|_DEFERRED to release VnRnMn
  OPEN_DATE:  yyyy-mm-dd       (date problem reported)
 CLOSE_DATE:  yyyy-mm-dd       (date problem resolved/closed)
 SERVICE(S):  ALL|GATE|LLC-n|PCNE|QLLC|TCPIP|TRACE|_CALLOUT... (component)
  MANDATORY:  NO|YES|_RECOMMENDED             (maintenance required?)
 ORIGIN/REF:  vrm_cid|rid,APARREF|N/A         (origin REF ID or REF APAR)
  PTF_CLASS:  N/A|TO-BE-DETERMINED|ENHANCEMENT-APAR|STANDARD-APAR
+  PTF_TYPE:  N/A|REFRESH|DEFERRED|CIRCUMVENTION
|  PTF_TYPE:  (OBJ) HNASOBJX and/or (SRC) HNASMACX
    PTF_LOC:  FTP Server Directory /hnas_maint/hnasvrmm/apars/
              (Complete non-SMP/E FIX is contained in the @apar.ZIP file)
              SMP/E PTFs are provided via user request because the
              Comm-Pro supplied MCS is unique to each customer.
   COREQ(S):  APARID|MAINT_LVL|N/A  (co-requisites)
+ PREREQ(S):  APARID|MAINT_LVL|N/A  (pre-requisites)
| PREREQ(S):  Distribution dated after: yyyy-mm-dd
|             With APARs: vrmnnnn and vrmnnnn applied.
  OBJECT(S):  N/A|member-names,
  SOURCE(S):  N/A|member-names,

    PROBLEM:  <one sentence problem assessment>
              Hung resource...|NASHALT at location...|Session Clears...

DESCRIPTION:  <complete problem description>

   SOLUTION:  <solution description>

CIRCUMVENTION:  <circumvention description>
                <provided in cases where a non-ptf circumvention such as
                 a configuration/coding change can eliminate the problem>

 APPLY_INFO:  See Chapter 6 (Product Maintenance Installation
              section) from the HNAS Guide and Reference Manual
              for instructions on how to install PTF's (Object and
              Source) or Refresh/Upgrade maintenance.

              Corrective logic included in distributions created
              after CLOSE_DATE.  Otherwise, apply maintenance as
              directed in the APPLY_INFO (PTF).
```

APAR Summaries / Maintenance Summaries / Memo Notices

For printed versions of the HNAS manual, this section is provided as a convenient place to locate HNAS APAR Reference and Maintenance Summary memo activity for future reference. Up-to-date copies of these maintenance summaries are available on our Web site and FTP Server.

For users of online PDF or HTML documentation, we recommend that you locate a copy of the respective HNAS V*n*R*n*M*n* APAR Reference and Maintenance Summaries files that you periodically download from our Web site or FTP Server into the *hlq*.*vrmnnnn*.HNASCNTL library to document product maintenance for your respective HNAS product release. Please refer to the Comm-Pro X.25 Host NAS Product Maintenance and Support Information web site link located under contact information for up-to-date APAR and Custom Enhancement activity.

Product Maintenance EOMDATE Authorization Key Extension

Commencing with the V2R4M0 release of the HNAS product (under maintenance APAR 2400095), we now provide a means of extending the End of Maintenance/Use Date (EOMDATE) without the requirement to install a product Refresh or Upgrade eDistribution. This is accomplished through the use of an EOMKEY= value that is passed to HNAS in the EXEC PARM= operand or via the MMEM console command. The EOMKEY= value is an encoded 16 decimal digit key that HNAS uses to extend the EOMDATE in your existing HNAS system.

EOMKEY order fulfillment processing for EOMDATE extension is supplied by your HNAS support provider. The keys are generated as renewals orders are processed for HNAS Maintenance/Use licenses.

APPENDIX A

X.3 PAD Parameters

This page left intentionally blank.

Appendix A - X.3 PAD Parameters Summary

Following is a list of standard X.3 PAD Parameters supported by most PAD providers. These parameters control the behavior between the terminal, local PAD and remote PAD. When a HNAS connection is established for a PAD type session, either default X.3 PADPARM values or user defined PADPARM=values will be exchanged using the X.29 protocol.

Most of these values can be coded in the HNAS PADPARM operand of the REMOTE Definition Statement. The X.3 values can also be controlled by the user manually from their X.28 session. This list is provided as a quick reference, we suggest that you refer to your PAD or Network provider for a detailed description regarding the appropriate use of these parameters as supported by their equipment.

For more information HNAS X.3 defaults, see CHAPTER 4 PADPARM operand description.

NUM	VAL	Function
1		**Controls escape to command mode.**
	0	Prevent DLE character from signalling escape.
	1	Allow DLE character to signal escape so that the user can modify PAD parameters.
2		**Controls echoplexing.**
	0	Prevent echoing of terminal input.
	1	Allow echoing of terminal input.
3		**Controls data forwarding.**
	0	Prevent control characters from forwarding terminal input.
	1	Allow any alphanumeric character to forward input.
	2	Allow CR (RETURN) character to forward input.
	4	Allow ESCAPE, BEL, ENQ or ACK character to forward input.
	8	Allow DEL, CAN or DC2 character to forward input.
	16	Allow ETX or EOT character to forward input.
	32	Allow HT, LT, VT or FF character to forward input.

X.3 PAD Parameters Summary

NUM	VAL	Function
	64	Allow ALL other characters in ASCII chart to forward input.
	126	Allow any control character to forward input.
4		**Controls time forwarding.**
	0	Prevent timer from forwarding input.
	1 - 255	Allow terminal input to be forwarded every 1/20th seconds
5		**Controls auxiliary device control capability.**
	0	Prevent auxiliary device from being controlled.
	1	Allow auxiliary device to be controlled with XON/XOFF.
6		**Controls network message suppression.**
	0	Prevent network messages from being sent to user.
	1	Allow network messages to be sent to user.
7		**Controls terminal break key (BRK/BREAK) mapping.**
	0	Prevent break key from signalling any function.
	1	Allow BRK to signal Interrupt Indication.
	2	Allow BRK to signal Reset Indication.
	8	Allow BRK to signal escape to command mode.
	15	Allow BRK to signal attention sequence not supported by XTP protocol).
8		**Controls discard output.**
		This parameter works in conjunction with parameter 7 (BREAK) and is controlled by the PAD process. **Do not** code parameter 8 in PAD-PARM.

NUM	VAL	Function
9		**Controls timefill idle character transmission.**
	0	Prevent idle characters from being transmitted after a CR is received from terminal.
	1 - 255	Allow n-idle characters to be transmitted after a CR is received from terminal.
10		**Controls line folding when input exceeds line size.**
	0	Prevent line folding (terminal may wrap data).
	1 - 255	Provide newline operation after n-characters are received from terminal.
11		**Controls baud rate values indication (in bits per second).**
	n = bps	10=50, 5=75, 9=100, 0=110, 1=134.5, 6=150, 8=200, 2=300, 4=600, 3=1200, 7=1800, 11=75/1200, 12=2400, 13=4800, 14=9600, 15=19200, 16=48000, 17=56000 or 18=64000.
12		**Controls network output flow control capability.**
	0	Prevent terminal operator from flow controlling.
	1	Allow terminal operator to flow control network output with XON/XOFF.
13		**Controls PAD line feed insertion capability.**
	0	Prevent line feed character insertion.
	1	Allow line feed insertion after transmitting a return (CR) to the terminal.
	2	Allow line feed insertion after echoing a return (CR) to the terminal.
	4	Allow line feed insertion after echoing a return (CR) to the remote host.
14		**Controls line feed padding.**
15		**Controls local editing capability (PAD editing mode).**

X.3 PAD Parameters Summary

NUM	VAL	Function
	0	Prevent editing capabilities while in data transfer mode.
	1	Allow editing capabilities while in data transfer mode.
16		**Controls character delete editing function (PAD editing mode).**
	0 - 127	Select a single character from the ASCII character set to represent the delete character. Default is character 18 (Ctrl-R).
17		**Controls line delete editing function (PAD editing mode).**
	0 - 127	Select a single character from the ASCII character set to represent the delete character. Default is character 21 (Ctrl-U).
18		**Controls line display editing function (PAD editing mode).**
	0 - 127	Select a single character from the ASCII character set to represent the display character. Default is character ?? (Ctrl-?).

APPENDIX B

Host NAS Configuration Examples

This page left intentionally blank.

Appendix B - Host NAS Configuration Examples

The HNAS initialization process creates the resources HNAS needs by interpreting a Configuration Data File (CDF). The following sample configurations and the resulting HNAS SYSPRINT log files illustrate typical HNAS definition statement usage and the processing that is performed during HNAS initialization.

HNAS Configuration Data File Statements for Cisco XOT Environment

```
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        HNAS V2R2M0 - SAMPLE CONFIGURATION
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         BUILD BFRLMT=1024             ; BUFFER POOL REQUIREMENT
*              BFRSIZ=180              ; FOR PKTSIZ=128 (1-BFR/PKT)
               BFRSIZ=564              ; FOR PKTSIZ=512 (1-BFR/PKT)
*              BFRSIZ=1076             ; FOR PKTSIZ=1024 (1-BFR/PKT)
               CONLMT=1                ; REMOTE CONSOLE COUNT
               CONPSWD=PASSWORD        ; CONSOLE PASSWORD
               CONPRMT='HNAS220>'      ; CONSOLE INPUT PROMPT
               NASNAME=HNAS220         ; OUR NAME
               PRTLMT=999999           ; TOTAL NUMBER PRINT RECORDS
*              MSGLMT=245              ; TCPIP MSG LIMIT (USE DEFAULT)
               TCPNAME=TCPIP           ; TCP/IP ADDRESS SPACE NAME
               TRCLMT=16384            ; TRACE TABLE REQUIREMENT
*              VCLMT=50                ; VC LIMIT (USE DEFAULT)
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOCAL XOT SERVER
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTSRVR  LOCAL TYPE=XOT                ; SERVER TYPE
               IPADDR=10.117.56.221    ; SERVER IP ADDRESS MVS
               PORT=1998               ; SERVER PORT NUMBER
               RTEIN=(MCH1/001,        ; CALLED ADDRESS=001 FOR MCH1
                      MCH2/002,        ; CALLED ADDRESS=002 FOR MCH2
                      MCH4/004,        ; CALLED ADDRESS=004 FOR MCH4
                      MCHCONS)         ; ROUTE EVERYTHING ELSE TO CONS
               RTEOUT=XOTCNOT          ; ROUTE OUTBOUND CALLS
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        REMOTE XOT - CISCO ROUTER PORTS (INBOUND SOCKET POOL)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTCNIN  REMOTE TYPE=XOT               ; ROUTER TRANSPORT PROTOCOL
               INIT=ACTIVE             ; ACTIVATE AT STARTUP
               IPADDR=10.117.56.100    ; ROUTER IP ADDRESS 3640
               PORT=DYNAMIC            ; ROUTER INITS CONNECT
               VCLMT=30                ; ROUTER VC COUNT
               TAP=0                   ; ROUTER KEEP ALIVE OFF
```

```
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        REMOTE XOT - CISCO ROUTER PORTS (SHARED SOCKET POOL)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
XOTCNOT  REMOTE TYPE=XOT                ; ROUTER TRANSPORT PROTOCOL
               INIT=ACTIVE              ; ACTIVATE AT STARTUP
               IPADDR=10.117.56.100     ; ROUTER IP ADDRESS
               PORT=1998                ; HNAS INITS CONNECT
               VCLMT=30                 ; ROUTER VC COUNT
               TAP=0                    ; ROUTER KEEP ALIVE OFF
               HOME=XOTSRVR             ; HOME SERVER
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH #1 (EMULATES NPSI MCH GTMMCH1)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1     REMOTE TYPE=MCH                ; LOGICAL MCH
               MBITCHN=YES              ; INPUT IS ALWAYS FIC/MIC/LIC
               GATE=GENERAL             ; GATE
               OPTIONS=(REPDCEADDR,     ; ADD CALLING ADDRESS IN CALLOUT
                        STRIPRTEIN,     ; REMOVE CALLED ADDRESS IN CALLIN
                        MCHTMR=15)      ; LOOK AT SLUS EVERY 15-SECONDS
               CONNECT=YES              ; FAST CONNECT
               PVC=NONE                 ; NO PVC SUPPORT
               LUNAME=(LX25MCH1/IFN39BB/XU065/1/10)
               HOME=XOTSRVR             ; HOME SERVER
               DCEADDR=001              ; CALLING ADDRESS FOR CALLOUT
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH #2 (EMULATES NPSI MCH GTMMCH2)
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH2     REMOTE TYPE=MCH                ; LOGICAL MCH
               MBITCHN=YES              ; INPUT IS ALWAYS FIC/MIC/LIC
               GATE=GENERAL             ; GATE
               OPTIONS=(REPDCEADDR,     ; ADD CALLING ADDRESS IN CALLOUT
                        STRIPRTEIN,     ; REMOVE CALLED ADDRESS IN CALLIN
                        LCN0USED,       ; START VCID AT 0
                        MCHTMR=15)      ; LOOK AT SLUS EVERY 15-SECONDS
               CUD0=ALL                 ; ANY CUD WILL DO
               CONNECT=NO               ; NO FAST CONNECT
               SUBADDR=YES              ; UTILISATION DU SUBADRESSING
               PVC=NONE                 ; NO PVC SUPPORT
               LUNAME=LX25MCH2/IFN39BB  ; MCH LU NAME
               LLC4=(4)                 ; SUBD=4 SETS GATE
               LLC5=(5)                 ; SUBD=5 SETS PAD
               SVC4=(10,                ; GATE SLUS.
                     XCOM2000,
                     XCOM2001,
                     XCOM2002,
                     XCOM2003,
                     XCOM2004,
                     XCOM2005,
```

```
                     XCOM2006,
                     XCOM2007,
                     XCOM2008,
                     XCOM2009)
               SVC5=(10)                ; PAD SLUS.
               SYSL=(SUBD=5/0)          ; SI SUBADD=5 ON VA VERS CSFI
               APPLNAME=(IFN39BB)
               PAD=INTEG
               TRAN=EVEN
               DCEADDR=002              ; CALLING ADDRESS FOR CALLOUT
*              FAC=(420707              ; PKTSIZ=128 FOR CALLOUT
*                   430303)             ; W=3 FOR CALLOUT
*              VCLMT=40                 ; VC LIMIT (USE DEFAULT)
               HOME=XOTSRVR             ; HOME SERVER
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        LOGICAL MCH #4 - CFT
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH4     REMOTE TYPE=MCH                ; LOGICAL MCH
               MBITCHN=YES              ; INPUT IS ALWAYS FIC/MIC/LIC
               GATE=GENERAL             ; GATE
               OPTIONS=(REPDCEADDR,     ; ADD CALLING ADDRESS IN CALLOUT
                        STRIPRTEIN,     ; REMOVE CALLED ADDRESS IN CALLIN
                        MCHTMR=15)      ; LOOK AT SLUS EVERY 15-SECONDS
               CONNECT=NO               ; NO FAST CONNECT
               SUBADDR=YES              ; UTILISATION DU SUBADRESSING
               PVC=NONE                 ; NO PVC SUPPORT
               LUNAME=XLU0041/CFT2GATE  ; MCH LU NAME
               LLC4=(4)
               SVC4=(20)
               DCEADDR=004              ; CALLING ADDRESS FOR CALLOUT
*              FAC=(420707              ; PKTSIZ=128 FOR CALLOUT
*                   430303)             ; W=3 FOR CALLOUT
               HOME=XOTSRVR             ; HOME SERVER
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        THIS MCH IS FOR CONSOLE CONNECTION ONLY.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCHCONS  REMOTE TYPE=MCH                ; LOGICAL MCH
               PAD=INTEG                ; NO INTEGRATED PAD SUPPORT
               PADPARM=(1/1,2/1,3/2,4/0,5/2,7/21,12/1,13/7,21/0)
               PWPROT=YESWOCC           ; PASSWORD IS PROTECTED
               TRAN=SPACE               ; 7-BIT ASCII
               MBITCHN=YES              ; INPUT IS ALWAYS OIC
               GATE=GENERAL             ; GATE SUPPORT TO SET LLC
               CONNECT=NO               ; NO FAST CONNECT
               SUBADDR=YES              ; NO SUBADDR LLC SELECTION
               LLC0=NONE                ; NO LLC0
               LLC4=NONE                ; NO LLC4
               LLC5=(1)                 ; CONSOLE USES LLC5
               PVC=NONE                 ; NO PVC SUPPORT
*              SVC4=NONE                ; GATE USED FOR LLC ONLY
```

```
                SVC5=(1,                     ; ALLOW 1 CONSOLE
                      CONSSLU)               ; DUMMY SLU
                APPLNAME=(CONSOLE)           ; CONSOLE IS ONLY APPL
                SYSL=(SUBD=01/0,CUD0=00/0,CUD0=01/0,NULL/0)
                HOME=XOTSRVR                 ; HOME SERVER
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*       DELIMIT CDF.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        END                                  ; IGNORE ANYTHING THAT FOLLOWS
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*       SAVED STATEMENTS.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*       LOGICAL MCH #1 - CFT CSFI
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MCH1    REMOTE TYPE=MCH                      ; LOGICAL MCH
                MBITCHN=YES                  ; INPUT IS ALWAYS FIC/MIC/LIC
                GATE=GENERAL                 ; GATE
                OPTIONS=(REPDCEADDR,         ; ADD CALLING ADDRESS IN CALLOUT
                         STRIPRTEIN)         ; REMOVE CALLED ADDRESS IN CALLIN
                CUD0=(C4,11,22,33,44,55,66) ;  2*CTCP CSFI ET CFT2
                CTCP=(00,01,01,01,01,01,01) ;  2*CTCP CSFI ET CFT2
                CONNECT=NO                   ; NO FAST CONNECT
                SUBADDR=YES                  ; UTILISATION DU SUBADRESSING
                DCEADDR=001                  ; ADD CALLING ADD IN CALL OUT
                PVC=NONE                     ; NO PVC SUPPORT
                LUNAME=(XLU0011,             ; MCH LU NAME
                        XLU0012)
                LLC0=NONE
                LLC4=(4)
                LLC5=(5)                     ; MINITEL ACCES NON GATE
                SVC0=NONE                    ; NO PCNE
                SVC4=(20)
                SVC5=(20)                    ; PAD SLU
                SYSL=(SUBD=5/0)              ; SI SUBADD=5 ON VA VERS CFT2GATE
                APPLNAME=CFT2GATE
                PAD=INTEG
                TRAN=EVEN
                FAC=(420707                  ; PKTSIZ=128
                     430303)                 ; W=3
                VCLMT=40                     ; ROUTER VC COUNT
*
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*       MCH EXTENSION(S) FOR ADDITIONAL CUD/FAC/DCEADDR VALUES.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MXT1    REMOTE TYPE=MXT                      ; LOGICAL MCH EXTENSION
                DCEADDR=NONE                 ; MCH DCE ADDRESS
```

```
CUD=01791020            ; PAD CALLOUT USER DATA
FAC=(0100               ; REVERSE CHARGING
     420707             ; PKTSIZ=128
     430404)            ; W=4
```

The CDF above is referenced by the //CONFIG DDNAME in the following HNAS job step.

HNAS Runtime Output from CDF for Cisco XOT Environment

Note: The HNAS runtime output that used to be provided in this Appendix has been removed. If you would like an example of a typical HNAS run, we can provide you with an up-to-date HNAS log via FTP or as an e-mail attachement.

HNAS Application Major Node File (AMNF) for Cisco XOT Environment

The following sample AMNF was created by the HNAS FASTRUN process for the CDF described above. The AMNF is stowed as a sequential file identified by the //MAJNODE DD statement. The //MAJNODE DD statement, like the //CONFIG DD statement, may point at a member of a partitioned dataset.

```
HNAS220  VBUILD TYPE=APPL
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCH1 OPERAND LUNAME=
LX25MCH1 APPL  DLOGMOD=INTERACT,EAS=1
XU065001 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065002 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065003 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065004 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065005 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065006 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065007 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065008 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU065009 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XU06500A APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCH2 OPERAND LUNAME=
LX25MCH2 APPL  DLOGMOD=INTERACT,EAS=1
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCH2 OPERAND SVC4=
XCOM2000 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XCOM2001 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XCOM2002 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
XCOM2003 APPL  DLOGMOD=INTERACT,                                       *
               EAS=1
```

```
XCOM2004 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
XCOM2005 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
XCOM2006 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
XCOM2007 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
XCOM2008 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
XCOM2009 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCH2 OPERAND SVC5=
MCH25001 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25002 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25003 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25004 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25005 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25006 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25007 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25008 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH25009 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH2500A APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCH4 OPERAND LUNAME=
XLU0041  APPL  DLOGMOD=INTERACT,EAS=1
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCH4 OPERAND SVC4=
MCH44001 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH44002 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH44003 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
MCH44004 APPL  DLOGMOD=INTERACT,                                        *
               EAS=1
```

```
MCH44005 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44006 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44007 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44008 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44009 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH4400A APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH4400B APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH4400C APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH4400D APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH4400E APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH4400F APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44010 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44011 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44012 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44013 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
MCH44014 APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
*
*
*        THE FOLLOWING SLUS ARE FOR REMOTE MCHCONS OPERAND SVC5=
CONSSLU  APPL  DLOGMOD=INTERACT,                                        *
                EAS=1
```

APPENDIX C

Host NAS Router Checklist

This page left intentionally blank.

Appendix C - Router Checklist

HNAS operates with the Cisco (XOT) and/or IBM (XTP) family of Internet routers. Each type of router must be properly configured to communicate with HNAS. This appendix attempts to generally address a variety of router topics such as; router physical configuration requirements, IOS software configuration requirements (including sample XOT protocol level configuration), basic post installation generation testing, remote access, diagnostics overview with some very useful debug/show tools with sample displays.

Please refer to **Chapter 3** in the **'HNASBook** for samples of various LLC configuration options for callin, callout and PVC environments while **Chapter 4** in the **'HNASBook** contains the various HNAS CDF parameters.

Cisco Router HNAS Documentation Section References

- **Router Environment Considerations (Cisco)** information is located in **Chapter 2**, of the **HNAS Guide and Reference Manual**. Please be sure and read this section.

- **Router Checklist**, providing Cisco router configuration, debug and display information is located in **Appendix C** of the **HNAS Guide and Reference Manual**.

- **Cisco Message Associations and HNAS Event Relationships**, providing Cisco specific or related alert and error message information is located in the **Messages and Codes Debugging Guide**.

Cisco Router XOT Environment

Cisco routers are the most widely used XOT routers in the industry. This router checklist overview contains some basic information that will assist you in supporting these routers in a HNAS environment. Please refer to your Cisco documentation for additional information on any of the Cisco processes identified in this section.

Cisco Router Physical Configuration

Hardware

You will require a serial interface port with an appropriate interface cable for each of the physical X.25 links that you plan to support. We use a WIC-2A/S Async/Sync 2 port WAN card in our 2621 that is one of the Cisco routers used for testing in our lab.

Storage/Memory

Ensure that your router has the required storage/memory for your IOS level using XOT. We run with 16/64 (16MB FLASH and 64MB DRAM) in our 2621 router. These memory sizes allow us to configure more robust Enterprise IOS levels.

Router Checklist

Software

Your router's IOS software must support XOT (also referred to as tunneling). We suggest that you refer to the Cisco software support matrix for confirmation.

Network

Ethernet and Token Ring are the two primary LAN protocols in use for HNAS connectivity. We use one of the 2621 10/100 Ethernet Adapters for HNAS access.

Remote Access

We suggest that you set up temporary dial-in access (with modem) using the on-board AUX port (TTY/VT100), direct on-board CON (TTY/VT100) access or a PPP port with TELNET access. This is so we can remotely access the router during initial product implementation. The access can be disabled or removed once product certification is completed. The remote access is not required as long as a router technician is available to provide router event or trace information should that become necessary.

Cisco Router XOT Software Configuration

The following configuration example provides an overview of the Cisco router configuration requirement for XOT support. It should give your router technician the information he or she needs to define the basic XOT environment for eventual HNAS X.25 access.

In this example, it is assumed that a Cisco router will be a local peer router with two X.25 links that are modem attached directly to the router. The remote X.25 originating addresses for interface serial0/1 X.25 link are 10360001 and 10360002 while the interface serial0/0 X.25 link is defined for TRANSPAC access. The X.25 destination address associated with the router's XOT interface serial0/1 is 203600nn (nn varies depending upon the assigned SUBD value, if provided). The X.25 destination address associated with the router's XOT interface serial0/1 is address 001 which is associated with the MCH (in this example interface serial0/1 is associated with MCH1). The IP address of 10.117.56.221 is the address of the host's TCP/IP stack.

1. Enable X.25 switching (XOT support).
   (Cisco and Comm-Pro recommend that ‘service tcp-keepalives’ be enabled)

   ```
   x25 routing
   service tcp-keepalives-in
   service tcp-keepalives-out
   ```

   Note - add **xot-keepalive-period 10 xot-keepalive-tries 3** parameters (for your specific environment requirements to the ‘x25 route ...’ strings to set specific tcp-keepalive period and tries values.

2. Define route types for the interface and XOT routing.

   ```
   x25 route ^10360001$ interface Serial0/1
   ```

```
x25 route ^10360002$ interface Serial0/1
!
x25 route ^203600 xot 10.117.56.221
!
x25 route source 001 interface Serial0/0
x25 route .* input-interface serial0/0 substitute-dest 001\0  xot 10.117.56.221
```

Note - If you are using a network provider such as TRANSPAC (or use an X.25 PAD or Switch) that doesn't provide called DTE addresses we suggest that you initially include x25 route definitions that will cause all inbound calls on the routers serial interfaces to be routed directly to the XOT host. You may also need to enable the '**x25 pad-access**' parameter on the serial x25 interface for these network types. This is done to eliminate the potential condition where there is no called address in the call request packet arriving over the x25 serial interface for the router to associate with an XOT host destination so connection to the router console command prompt> is established.

3. Define the Physical X.25 Interface.

The interface should have the basic LAPB and X.25 parameters defined that correspond to the new X.25 links. DCE|DTE, V35|RS232|etc., SPEED=, FRAME WINDOWS_SIZE (K=7) PACKET WINDOW_SIZE (W=2), etc. The majority of defaults are correct. All serial LAPB, X.25 FRAME and PACKET values must conform to the X.25 network's requirements. Some networks requires that the HNAS REMOTE option **lcn0=used** be enabled because data packets support is required on logical channel number 0 for some older TRANSPAC implementations). Following is a sample of the serial interfaces:

```
!
interface Serial0/0
 description TRANSPAC_Link_MCHn
 no ip address
 no ip directed-broadcast
 encapsulation x25
 no ip mroute-cache
 x25 htc 10
! default window size is 2 (x25 win 2 and x25 wout 2
 x25 win 3
 x25 wout 3
 x25 suppress-calling-address
 x25 subscribe lc0 two-way          <- Required for some inbound/outbound calls
!                                      requiring LCN 0 data sessions.
 x25 pad-access                     <- Required for inbound calls without a called
!                                      address.
interface Serial0/1
 description Swift_Link_1_(10360001)
 no ip address
 no ip directed-broadcast
 encapsulation x25
 no ip mroute-cache
 x25 htc 10
! default packet size is 128 (x25 ips 128 and x25 ops 128)
```

```
 x25 ips 512
 x25 ops 512
 x25 accept-reverse
!
```

PVC's are also supported although not provided in the above sample. Please refer to Chapter 3 section 'Example Cisco Router XOT PVC Configuration' for additional information.

4. As with Cisco serial X.25 interfaces, some networks require that logical channel number 0 (LCN 0) support be enabled for data packet flow (normally LCN 0 is reserved for control packets while LCN 1 and above are used for data sessions). We have observed this requirement with some TRANSPAC networks running with host GATE callout applications (e.g., CSFI) although PCNE, QLLC and PAD environments may require this support as well because the LCN 0 option is an X.25 network requirements. The following optional definitions must be used to facilitate this function for inbound and outbound sessions.

   ```
   x25 subscribe lc0 incoming-only|two-way|outgoing-only
   ```

   Note that LCN0 may be reserved for SVC use only. Consequently, there cannot also be PVCs on an MCH link that uses LCN0 since PVCs must start at the lowest logical channel number. Additional commands for LCN0 are:

   ```
   show x25 vc 0
   clear x25 vc 0
   debug x25 vc 0
   ```

5. Optional definitions to allow TELNET sessions to access Cisco pad function for connection to HNAS via XOT. Can use to test HNAS console access. Note that sessions use VTY lines.

   ```
   service pad to-xot
   service pad from-xot
   x25 host cons 20360009
   x25 host tso 20360002
   ```

That's about it for a basic 26nn router configuration. As you can see, the definitions are fairly simple. If you provide your HNAS technical support services representative with the following information they can assist you in creating a sample HNAS Configuration Data File (CDF):

IP address of the router (Ethernet port or Loopback address) callout destination router)
IP address of the host TCP/IP stack
NPSI source (Major Node) with X.25 resources identified for the migration to HNAS
NPSI Switched Major Node definition file for NPSI resources required for the migration

Software (IOS) Hidden Commands

The Cisco IOS software has some “hidden commands” that provide expanded features for router IOS operations. The hidden commands provide various capabilities not provided in the standard Cisco product documentation. Our experience with these existing hidden commands is that they are generally provided by Cisco TAC in response to problem reports. Cisco TAC can provide you with the hidden command description and parameter settings for the following hidden commands that we have come across:

‘x25 subscribe lc0 incoming-only|two-way|outgoing-only’
‘x25 version 1980|1984|1988|1993’ (**Note:** This feature was integrated into Cisco IOS Release 12.3(9) as a standard command).

Note: HNAS was developed using the x25 version 1980 specification. Newer x25 version options are provided as required under specific HNAS CDF options.

Cisco Router XOT Configuration Test

Once the Cisco router is configured for XOT with the X.25 interfaces defined, we would like you to perform the following tests and supply us with the results:

1. After loading or activating the revised router configuration, issue '**show x25**' from the Cisco router console. The display output should indicate that the X.25 and XOT software is present with their respective versions.

2. If possible, attach an active/working X.25 link to the router interface and issue '**show interface**' from the Cisco router console. The display will provided interface status (modem signals, etc.), general statistics and interface configuration information. Confirm that the X.25 link comes up.

   The purpose of performing these actions is to ensure that the hardware and software are configured properly prior to any HNAS host application testing.

3. Prior to host application testing, be sure to activate event logging by issuing '**debug x25 event**' from the Cisco router console prior to enabling host application data flow. This will cause the router to generate several event alarms that we can review for call establishment and call clearing activity. Once debugging is completed please be sure and disable the router debug process by entering '**no debug x25**' to eliminate unnecessary router debugging CPU activity.

   Should additional information be required such as packet level data flow, you can issue '**debug x25 all**', but **be aware** that this additional event logging will consume more router CPU cycles.

Cisco Router Diagnostics (Debug/Show)

The Cisco debugging tools provide exceptional diagnostic information for the end point router's X25 Serial Interface tunneling to/from XOT Host Interfaces. This includes session connect, data transfer, connectivity transitions, and disconnect activity with the X25 Network on the Serial side and the HNAS XOT product on the TCP/IP Host Stack side.

Invaluable information is available when using various Cisco show commands such as: **show interface**, **show x25**, **show x25 xot** and **show x25 vc**. A brief description of these show commands and the content of the display output is provided in the samples available in this section.

Various levels of message and packet activity information for X25 or XOT specific events is generated using the '**debug x25 event**', '**debug x25 xot**', '**debug x25 all**', etc. router debug commands. The Cisco TCP/IP '**debug ip**' and '**debug ip tcp**' are not discussed in this section because the debug information generated by these command is rarely required for X25-over-TCP/IP debugging.

The Cisco Debug Command Reference documentation (along with the Internetwork Troubleshooting Guide) are invaluable debugging tools for use in researching router IOS system and debug message activity. We suggest that you consult these guide's as well as the 'Cisco Messages Related to HNAS' section of the 'HNAS Messages and Codes' manual for additional information.

In the following samples, only a single test session was active or used to generate connect and disconnect sequences. This was done so that the differences in trace debug output could easily be viewed and compared with each of the provided debug options. Refer to the Cisco debug documentation for information on interface and resource filtering to reduce debug message activity when multiple interfaces and sessions are active.

Please be sure and review the Cisco documentation **warning** information regarding potential impact on **network performance when debugging is employed.**

Note: The content provided in the following displays may not match the Cisco Router XOT Software Configuration samples provided earlier in this manual.

Show Interface Serial - Sample Show Interface Serial Display

Following is the sample output for the Cisco show interface command. In this sample we are viewing interface serial0/1 which is the same interface used for the sample debug x25 displays in this section. This display contains various configuration parameter setting as well as the link and protocol level state and several global counters with very useful information.

```
cp2621> show interface serial0/1
Serial0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Description: BBOX_PAD-8 8 Port 32 VC X.25 Pad (10360001)
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation X25, loopback not set
  X.25 DTE, address <none>, state R1, modulo 8, timer 0
      Defaults: idle VC timeout 60
        cisco encapsulation
        input/output window sizes 2/2, packet sizes 128/128
      Timers: T20 180, T21 200, T22 180, T23 180
      Channels: Incoming-only none, Two-way 1-20, Outgoing-only none
      RESTARTs 0/0 CALLs 4+0/13+1/13+0 DIAGs 0/0
  LAPB DTE, state CONNECT, modulo 8, k 7, N1 12056, N2 20
      T1 30000, T2 0, interface outage (partial T3) 0, T4 45
      VS 3, VR 0, tx NR 0, Remote VR 3, Retransmissions 0
      Queues: U/S frames 0, I frames 0, unack. 0, reTx 0
      IFRAMEs 989/1055 RNRs 0/0 REJs 0/0 SABM/Es 0/0 FRMRs 0/0 DISCs 0/0
  Last input 12w2d, output 00:00:07, output hang never
  Last clearing of "show interface" counters 7w3d
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     152362 packets input, 308511 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     152287 packets output, 374793 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
cp2621>
```

Show X25 - Sample Show X25 Display

Following is the sample output for the Cisco show x25 command. This command is useful in determining if the X25 and XOT features are defined in the router. The display also includes the X25 and XOT protocol software levels and Virtual Circuit counts and usage.

```
cp2621#: show x25
X.25 software, Version 3.0.0.
  5 configurations supporting 4 active contexts
  VCs allocated, freed and in use: 5305 - 5300 = 5
  VCs active and idle: 1, 4
XOT software, Version 2.0.0.
  1 configurations supporting 1 active contexts
  VCs allocated, freed and in use: 26 - 24 = 2 (1 idle)
  connections: 0 outgoing, 0 incoming, 1 active, 0 orphaned,
               0 idle; 1 total
  active VCs: 1, connected to 1 remote hosts
```

Show X25 VC - Sample Show X25 VC Display

Following is the sample output for the Cisco show x25 vc command. There is one connection on the X25 serial interface and one connection on the XOT interface for each connected user session. The display information is only available while user sessions are connected. There is some very useful information in this display output.

```
cp2621#: show x25 vc
SVC 1,  State: D1,  Interface: Serial0/1
  Started 00:01:24, last input 00:01:20, output 00:01:20
  Connects 1036000103 <--> 20360009 to
  XOT between 10.117.56.100, 13205 and 10.117.56.221, 1998
  Window size input: 2, output: 2
  Packet size input: 128, output: 128
  PS: 5  PR: 1  ACK: 1  Remote PR: 5  RCNT: 0  RNR: no
  P/D state timeouts: 0  timer (secs): 0
  data bytes 72/5 packets 5/1 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0
SVC 1,  State: D1,  Interface: [10.117.56.221,1998/10.117.56.100,13205]
  Started 00:01:24, last input 00:01:20, output 00:01:20
  Connects 1036000103 <--> 20360009 from Serial0/1 SVC 1
  Window size input: 2, output: 2
  Packet size input: 128, output: 128
  PS: 1  PR: 5  ACK: 5  Remote PR: 1  RCNT: 0  RNR: no
  P/D state timeouts: 0  timer (secs): 0
  data bytes 5/72 packets 1/5 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0
cp2621>
```

Sample XOT Debug Error Messages

Following are a few sample debug messages that are generated by the Cisco router (XOT debug enabled) when HNAS isn't active during a session connect attempt or when HNAS is abruptly terminated with an active XOT session:

[**hhh.hhh.hhh.hhh (pending)]: XOT open failed (connection refused by remote host)**

This message is issued after a Cisco router attempt's an XOT connection to HNAS but is unable to locate an open socket. This condition typically occurs when HNAS is inactive (no longer running) or when the IP or PORT addresses are incorrect.

The **hhh.hhh.hhh.hhh** value identifies the destination IP address where the XOT host services are running.

[**hhh.hhh.hhh.hhh,1998/rrr.rrr.rrr.rrr,pppp]: XOT receive error, End of data transfer**

This message is issued by the router when an XOT session is abruptly terminated due to a HNAS shutdown or abend condition. This occurs because the sockets aren't closed in an orderly fashion after a Cisco router attempt's an XOT connection to HNAS but is unable to locate an open socket. This condition typically occurs when HNAS is inactive (no longer running) or when the IP or PORT addresses are incorrect.

The **hhh.hhh.hhh.hhh,1998** values identify the destination IP address and XOT port address **(1998)** where the XOT host services are running while the **rrr.rrr.rrr.rrr,pppp** values identify the XOT router's IP address and port address.

Please refer to the **'HNAS Messages and Codes Debugging Guide'** section **'Cisco Messages Relating to HNAS Events'** for additional Cisco XOT router support information.

Refer to the Cisco documentation for a complete description of Cisco message formats and status or error messages generated by Cisco routers.

Debug X25 Event - Sample X25/XOT Session Event Messages

Following are sample X25 serial interface and XOT protocol level system messages generated by enabling the denoted debug command. This debug information is very useful in monitoring Serial X25 interface and XOT protocol call set-up and clear activity. No data packet or RR activity is included in this debug option.

```
cp2621#: debug x25 event
X.25 special event debugging is on
cp2621#:

dttm.759: Serial0/1: X.25 I R1 Call (18) 8 lci 1
dttm.759:   From (10): 1036000103 To (8): 20360009
dttm.759:   Facilities: (0)
dttm.759:   Call User Data (4): 0x01000000 (pad)
dttm.775: [10.117.56.221,1998/10.117.56.100,13201]:
            XOT O P2 Call (24) 8 lci 1
dttm.775:     From (10): 1036000103 To (8): 20360009
dttm.775:     Facilities: (6)
dttm.775:       Packet sizes: 128 128
dttm.775:       Window sizes: 2 2
dttm.775:     Call User Data (4): 0x01000000 (pad)
dttm.815: [10.117.56.221,1998/10.117.56.100,13201]:
            XOT I P2 Call Confirm (3) 8 lci 1
dttm.815: Serial0/1: X.25 O R1 Call Confirm (3) 8 lci 1
dttm.603: [10.117.56.221,1998/10.117.56.100,13201]:
            XOT I P4 Clear (5) 8 lci 1
dttm.603:     Cause 0, Diag 140 (DTE originated/Unknown diagnostic)
dttm.603: Serial0/1: X.25 O R1 Clear (5) 8 lci 1
dttm.607:     Cause 0, Diag 140 (DTE originated/Unknown diagnostic)
dttm.655: Serial0/1: X.25 I R1 Clear Confirm (3) 8 lci 1
dttm.655: <detached>: XOT O P7 Clear Confirm (3) 8 lci 1
```

Note: The **dttm** (datetime field was removed to reduce message length. Only the **msec** field is present (Month dd hh:mm:ss.msec)

Debug X25 XOT - Sample XOT Session Messages

Following are sample XOT protocol system messages generated by enabling the denoted debug command. This is very useful when you want to view only the XOT protocol activity. XOT data packet and RR activity are included in this debug option.

```
cp2621#: debug x25 xot
X.25 packet debugging is on
X.25 debug output restricted to protocol XOT
cp2621#:

dttm.299: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT O P2 Call (24) 8 lci 1
dttm.303:     From (10): 1036000103 To (8): 20360009
dttm.303:     Facilities: (6)
dttm.303:       Packet sizes: 128 128
dttm.303:       Window sizes: 2 2
dttm.303:     Call User Data (4): 0x01000000 (pad)
dttm.339: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I P2 Call Confirm (3) 8 lci 1
dttm.343: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I D1 Data (12) Q 8 lci 1 PS 0 PR 0
dttm.347: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I D1 Data (39) 8 lci 1 PS 1 PR 0
dttm.351: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT O D1 RR (3) 8 lci 1 PR 2
dttm.371: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I D1 Data (6) Q 8 lci 1 PS 2 PR 0
dttm.587: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT O D1 Data (8) 8 lci 1 PS 0 PR 3
dttm.611: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I D1 Data (6) Q 8 lci 1 PS 3 PR 1
dttm.611: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I D1 Data (24) 8 lci 1 PS 4 PR 1
dttm.611: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT O D1 RR (3) 8 lci 1 PR 5
dttm.807: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT O D1 Data (6) 8 lci 1 PS 1 PR 5
dttm.079: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I D1 Data (26) 8 lci 1 PS 5 PR 2
dttm.087: [10.117.56.221,1998/10.117.56.100,13199]:
           XOT I P4 Clear (5) 8 lci 1
dttm.087:     Cause 0, Diag 140 (DTE originated/Unknown diagnostic)
dttm.139: <detached>: XOT O P7 Clear Confirm (3) 8 lci 1
```

Note: The **dttm** (datetime field was removed to reduce message length. Only the **msec** field is present (Month dd hh:mm:ss.msec)

Debug X25 Int - Sample X25 Serial Int Session Messages

Following are sample X25 Interface system messages generated by enabling the denoted debug command. This is very useful when you want to view only the serial x25 interface activity. X25 data packet and RR activity are included in this debug option.

```
cp2621#: debug x25 interface serial0/1
X.25 packet debugging is on
X.25 debug output restricted to interface Serial0/1
cp2621#:

dttm.163: Serial0/1: X.25 I R1 Call (18) 8 lci 1
dttm.163:   From (10): 1036000103 To (8): 20360009
dttm.163:   Facilities: (0)
dttm.163:   Call User Data (4): 0x01000000 (pad)
dttm.219: Serial0/1: X.25 O R1 Call Confirm (3) 8 lci 1
dttm.219: Serial0/1: X.25 O D1 Data (12) Q 8 lci 1 PS 0 PR 0
dttm.283: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 1
dttm.375: Serial0/1: X.25 O D1 Data (39) 8 lci 1 PS 1 PR 0
dttm.387: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 2
dttm.395: Serial0/1: X.25 O D1 Data (6) Q 8 lci 1 PS 2 PR 0
dttm.427: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 3
dttm.891: Serial0/1: X.25 I D1 Data (8) 8 lci 1 PS 0 PR 3
dttm.911: Serial0/1: X.25 O D1 Data (6) Q 8 lci 1 PS 3 PR 1
dttm.915: Serial0/1: X.25 O D1 Data (24) 8 lci 1 PS 4 PR 1
dttm.947: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 4
dttm.947: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 5
dttm.671: Serial0/1: X.25 I D1 Data (6) 8 lci 1 PS 1 PR 5
dttm.459: Serial0/1: X.25 O D1 Data (26) 8 lci 1 PS 5 PR 2
dttm.467: Serial0/1: X.25 O R1 Clear (5) 8 lci 1
dttm.467:   Cause 0, Diag 140 (DTE originated/Unknown diagnostic)
dttm.471: Serial0/1: X.25 I P6 RR (3) 8 lci 1 PR 6
dttm.523: Serial0/1: X.25 I R1 Clear Confirm (3) 8 lci 1
```

Note: The **dttm** (datetime field was removed to reduce message length. Only the **msec** field is present (Month dd hh:mm:ss.msec)

Debug X25 All - Sample X25/XOT Session Messages

Following are sample X25 serial interface and XOT protocol level system messages generated by enabling the denoted debug command. This debug information is very useful in monitoring Serial X25 interface and XOT protocol data packet and RR activity.

Warning: The '**all**' option enables global event logging on ALL X25 serial interfaces and XOT protocol IP addresses for the router being debugged. This may severely impact network performance when several resources are active. Only one session was in use during this debug process.

```
cp2621#: debug x25 all
X.25 packet debugging is on
cp2621#:

dttm.307: Serial0/1: X.25 I R1 Call (18) 8 lci 1
dttm.307:   From (10): 1036000103 To (8): 20360009
dttm.311:   Facilities: (0)
dttm.311:   Call User Data (4): 0x01000000 (pad)
dttm.323: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT O P2 Call (24) 8 lci 1
dttm.323:     From (10): 1036000103 To (8): 20360009
dttm.323:     Facilities: (6)
dttm.323:       Packet sizes: 128 128
dttm.323:       Window sizes: 2 2
dttm.323:     Call User Data (4): 0x01000000 (pad)
dttm.363: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I P2 Call Confirm (3) 8 lci 1
dttm.363: Serial0/1: X.25 O R1 Call Confirm (3) 8 lci 1
dttm.363: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I D1 Data (12) Q 8 lci 1 PS 0 PR 0
dttm.363: Serial0/1: X.25 O D1 Data (12) Q 8 lci 1 PS 0 PR 0
dttm.371: [10.117.56.221,1998/10.117.56.100,13203]:
             XOT I D1 Data (39) 8 lci 1 PS 1 PR 0
dttm.371: Serial0/1: X.25 O D1 Data (39) 8 lci 1 PS 1 PR 0
dttm.371: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT O D1 RR (3) 8 lci 1 PR 2
dttm.395: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I D1 Data (6) Q 8 lci 1 PS 2 PR 0
dttm.431: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 2
dttm.431: Serial0/1: X.25 O D1 Data (6) Q 8 lci 1 PS 2 PR 0
dttm.467: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 3
dttm.311: Serial0/1: X.25 I D1 Data (8) 8 lci 1 PS 0 PR 3
dttm.311: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT O D1 Data (8) 8 lci 1 PS 0 PR 3
dttm.335: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I D1 Data (6) Q 8 lci 1 PS 3 PR 1
dttm.335: Serial0/1: X.25 O D1 Data (6) Q 8 lci 1 PS 3 PR 1
dttm.335: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I D1 Data (24) 8 lci 1 PS 4 PR 1
dttm.335: Serial0/1: X.25 O D1 Data (24) 8 lci 1 PS 4 PR 1
dttm.335: [10.117.56.221,1998/10.117.56.100,13203]:
```

```
           XOT O D1 RR (3) 8 lci 1 PR 5
dttm.367: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 4
dttm.371: Serial0/1: X.25 I D1 RR (3) 8 lci 1 PR 5
dttm.463: Serial0/1: X.25 I D1 Data (6) 8 lci 1 PS 1 PR 5
dttm.463: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT O D1 Data (6) 8 lci 1 PS 1 PR 5
dttm.299: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I D1 Data (26) 8 lci 1 PS 5 PR 2
dttm.299: Serial0/1: X.25 O D1 Data (26) 8 lci 1 PS 5 PR 2
dttm.307: [10.117.56.221,1998/10.117.56.100,13203]:
            XOT I P4 Clear (5) 8 lci 1
dttm.307:   Cause 0, Diag 140 (DTE originated/Unknown diagnostic)
dttm.307: Serial0/1: X.25 O R1 Clear (5) 8 lci 1
dttm.307:   Cause 0, Diag 140 (DTE originated/Unknown diagnostic)
dttm.311: Serial0/1: X.25 I P6 RR (3) 8 lci 1 PR 6
dttm.359: Serial0/1: X.25 I R1 Clear Confirm (3) 8 lci 1
dttm.363: <detached>: XOT O P7 Clear Confirm (3) 8 lci 1
```

Note: The **dttm** (datetime field was removed to reduce message length. Only the **msec** field is present (Month dd hh:mm:ss.msec)

Show Translate - Sample Show Translate Display (telnet to XOT)

The Cisco translate support can be used for remote HNAS console access or access to LLC0 (PCNE) or LLC5 (PAD) applications using a telnet initiated session. No X.25 serial interfaces are required on the router to access the XOT host using translation.

```
translate tcp 10.117.56.123 binary x25 20360009 profile hnasfac1
x25 route ^203600 xot 10.117.56.221
x29 profile hnasfac11 2:1 3:2 4:0 5:2 7:21 12:1 13:7 21:0 <-test values
```

Following is the sample output for the Cisco show translate command. Each inbound telnet connection received by the router is associated with an XOT host and assigned an XOT vc session.

```
cp2621#: show translate

Translate From: TCP 10.117.56.240 Port 23 Printer
          To:   X25 30360009
          Quiet
          0/0 users active, 0 peak, 0 total, 0 failures

Translate From: TCP 10.117.56.121 Port 23 Binary
          To:   X25 20360009
          Quiet
          0/0 users active, 1 peak, 1 total, 0 failures
Translate From: TCP 10.117.56.123 Port 23 Binary
          To:   X25 20360009 Profile hnasfac1
          Quiet
          1/0 users active, 1 peak, 12 total, 5 failures

cp2621#:
```

IBM Router XTP Environment

The router checklist overview content isn't being provided for IBM XTP routers due to their withdrawal from the general networking marketplace. Most in-house IBM router networking support representatives should be able to define and enable the IBM XTP environment for eventual HNAS X.25 access. Any specific questions can be presented directly to your HNAS technical support representative and every effort will be made to provide a response in a timely manner.

Router Checklist

APPENDIX D

Summary of Changes & New Features

This page left intentionally blank.

HNAS Summary of Changes & New Features (by VnRnMn level)

This Summary of Changes & New Features List by Release section serves as a quick reference to identify product features and enhancements as provided for recent HNAS VnRnMn release levels. Please refer to the specific ‘HNAS VnRnMn - New Features Overview’ section for a description of the summary items.

Additional information concerning operational changes for HNAS VnRnMn product refresh or upgrade activity can be found in Chapter 5 (Migration section) of the HNAS Guide and Reference Manual.

This page left intentionally blank.

HNAS V2R4M0 Release Summary:

Functional Enhancements

- Improvements in product installation for organizations operating in **non-SMP/E** environments
- **SMP/E** installation and maintenance is now fully integrated into the standard documentation
- **ISARX25 Datafono** support integrated into the standard HNAS product (subscription option)
- **BUILD CONCMDQ=*ddname* (CMDLIST)** support allows console command list and control for various activities
- **BUILD PRTSWLST=(*options*)** provides improved SYSPRINT automation and controls
- **PARM='EOTKEY=*dd*...*dd*'** start parameter allows the TRIAL PERIOD EXPIRATION DATE (**EOTDATE**) for a trial distribution to be extended using the key provided by the ***dd*...*dd*** digits
- **PARM='EOMKEY=*dd*...*dd*'** start parameter allows the MAINTENANCE/USE ANNIVERSARY DATE (**EOMDATE**) for a permanent distribution to be extended using the key provided by the ***dd*...*dd*** digits
- **PARM='EOMKEY=*dd*...*dd*'** start parameter also allows a trial distribution to be converted to a permanent distribution dynamically using the key provided by the ***dd*...*dd*** digits
- **PARM='PARMFILE=*pfddname*'** start parameter allows all or some HNAS start parameters to be provided in a sequential file
- **REMOTE TYPE=DMY** added permitting future expansion of some console command parms

Configuration Enhancements

- **ALRMFLTR=** table has been expanded from 16 to 128 entries allowing fine tuning of filters
- **ALRMFLTR=** now supports (FC) and (FU) message ID suffixes to allow forced conditional (subject to SHOW state) and forced unconditional (SHOW state ignored) SYSCONS display
- **CONCMDQ=** now includes the new **EXEC *ddname*** command list support
- **CONCMDQ=*ddname*** can now be specified to provide a dataset command list (provides consistency with the SCHEDULE=*ddname* operand)
- **EXEC=** operand has been added as an alias for CONCMDQ= (provides name symmetry with the EXEC command like the SCHEDULE= operand and SCHEDULE command)
- **EXEC=** operand has been modified to decode the new NOPURGEONERROR and PURGEONERROR keywords
- **FASTRUN AMNF** generation support now propagates VTAM parameters from the MCH to all GATE control session SLU APPL statements for the MCH
- **FASTRUN AMNF** generation support now propagates VTAM parameters from the MXT associated with a PVC SLU to the SLU APPL statement (MCH VTAM parameters used when MXT not coded)
- **INIT={ACTIVE|IDLE}** is now valid for TYPE=MCH REMOTE definition statements
- **INIT={ONLINE|OFFLINE}** is now accepted as alternate values for ACTIVE|IDLE, respectively
- **LUNAME=** operand has been modified to accept a **GENTYPE** value which is used to control hex versus decimal SLU name generation: LUNAME=(*sluname*-{A|I}/*pluname*/*pfxlu*-**{H|D**}/*sfxst*/*cnt*,...)
- **OPTIONS=ALRMSGTXT={LONG|SHORT}** now supported on BUILD
- **OPTIONS=CUD0SELECTSLU** (CUD0 LU selection) now supported for the TYPE=MCH|MXT REMOTE
- **OPTIONS=DELAYBINDRESP** now supported for the TYPE=MCH REMOTE
- **OPTIONS=IDTST** (Datafono M message validation) now supported for TYPE=DFX REMOTE
- **OPTIONS=NOCLOSEONTAPFAILURE** now supported for the TYPE=XOT|XTP REMOTE
- **OPTIONS=PVCRECONTMR=*secs***, **PVCSETUPTMR=*secs*** and **PVCSETUPREJ=*code*** now supported for the TYPE=MCH|MXT REMOTE
- **OPTIONS=RESETINO** now supported for the TYPE=MCH REMOTE
- **OPTIONS=SVCCALLTMR=*secs*** (T21=) now supported for the TYPE=MCH|MXT REMOTE
- **OPTIONS=(TIMESTAMP=*time-format*{+D})** on BUILD now provides SYSPRINT timestamp and datestamp controls
- **PARM=** (**EXEC**) operand now supports abbreviated start parameters reducing the likelihood that the operands 100 character limit will be exceeded

- **PARSE** definition statement support added permitting user defined CDF parsing characters
- **PRTSWLST**={LOOP|STOP,SWITCHAFTERINIT,SWITCHAT*time,*DYNAMIC=*x*} automation
- **PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)** operand added for heartbeat message support
- **REMOTE TYPE=DMY** permits predefined **PING (XOT)** parameters (all Call Request values)
- **REMOTE TYPE=SVC** added supporting new **SVC0|5=**{(...**<*rmtname*>**)} enhancement (see below)
- **RTEIN= SKIP** and **CLEAR** parameters available for filter list placeholders and call clear control
- **RTEIN=** and **RTEOUT=** operands now includes basic support for DTE address 'wildcards' filtering
- **RTEOUT=(*rmtname*/*dteaddr*/*desctxt*,...)** operand now accepts a descriptor text field
- **SCHEDULE={*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)}** operand support added to allow scheduled console commands
- ***sluname* INIT** value now allowed for the LUNAME=, PVC=, SVC0=, SVC4= and SVC5= operands: syntax is *sluname*-{**A**|**I**} where A is for active (online) and I is for idle (offline)
- **SOCLMT=*limit*** operand now generates an error message if the sum of the TYPE=XOT REMOTE VCLMT= values (for which the LOCAL is HOME) is greater than *limit*-2 (eliminates runtime ABEND)
- **SVC0|5={(*sluname*/{X*idnum*|*dteaddr*|<*rmtname*>}}** MXT|SVC per *dteaddr*, *dteaddr* via *mxtname*
- **TRCLMT=**count expanded from 32767 to 1048575 entries permitting extended tracing capabilities
- **TRCTRAP=ALRMLIST=(NAS*nnnni*/'*data*'/*data-offset*,...)** operand now accept ***data*** as quoted string
- See <u>Console and Trace Enhancements</u> for **PARM=** Start Parameter Changes & Improvements
- **SVC0|5={(*sluname+gluname*/...)}** generic SLU name support added to MCH

<u>Alert/Alarm Message and Clear/Reset Code Enhancements</u>

- **NASC*nnni*** console command response messages were added improving command decode status
- **NASC052W** console command response message issued if EXEC LIST is specified without command list (*ddname* or (*cmd*1,...,*cmd*n) <- <u>NASC052W retired by APAR 2400098</u>
- **NASC053W** console command response message issued if EXEC *ddname* identifies an empty command list file
- **NASC100E** console command response message has been modified to more completely identify why a command that operates on the LNM=, RNM=, LUNM= or ID= modifiers is aborted when none of these modifiers are specified.
- **NAS003*ni*** messages identify the type of HNAS shutdown as well as the actual end
- **NAS020*ni*** messages generated for **PRTSWLST**= SYSPRINT switching support
- **NAS011*ni*** messages generated for **SHOW {ON|OFF|ERR}** console command activity
- **NAS0130W** messages generated for **SHOW {MORE|LESS}** console command activity
- **NAS0210I** messages generated for **PRNT NEXTPRSW|SWITCHNOW** & **SWITCHAT*type*** activity
- **NAS021*ni*** messages generated for **PRNT {ON|OFF}** console command activity
- **NAS022*ni*** messages generated for **PRNT {OPEN|CLSOPN}** console command activity
- **NAS0230W** messages generated for **PRNT {DATE ON|OFF ...}** console command activity
- **NAS0299I** messages generated for **PULSE** heartbeat message
- **NAS0310W** messages generated for **VARY *mchname* |FORCE|OFF|ON** events
- **NAS1*n*11E** messages generated when incorrectly formatted table is specified for the USSTAB=, LOGTAB=, or OPTIONS=NRITAB= operand
- **NAS1*nnni*** configuration messages updated for new CDF parameters and options
- **NAS1154E** message generated when SCHEDULE= queue limit exceeded
- **NAS1301D** message is now issued when SLU initial value `(-{A|I})` is omitted
- **NAS1311S** message is now issued when the sum of the TYPE=XOT REMOTE VCLMT values (for which the LOCAL is HOME) is greater than SOCLMT=*limit* value minus 2 (NEEDED=*count* added)
- **NAS1391I** messages now issued to display MXT operand overrides
- **NAS2109S** message is now issued when an ***unexpected TCPIP interrupt*** is presented and the DBUG TCP start parameter is in effect
- **NAS2110S** message is now issued when an ***invalid TCPIP reply ID*** is detected and the DBUG TCP start parameter is in effect

- **`NAS2152E`** message is now issued when a CANCEL command timeout occurs
- **`NAS2502E`** message now displays CLOSEONTAPFAILURE when a router contact is lost and the CLOSEONTAPFAILURE option is in effect
- **`NAS2505E`** message was added and displays NOCLOSEONTAPFAILURE when a router contact is lost and the NOCLOSEONTAPFAILURE option is in effect
- **`NAS251nM`** messages that are issued for **MON TAP** now provide information similar to **PING** console command activity
- **`NAS2511M`** message that is issued for **MON TAP** now contains a retry count
- **`NAS261nM`** messages now issued for **PING** console command activity
- **`NAS261nM`** messages updated for **PING** console command to display a sequence number so that termination messages can be easily correlated to initiation messages
- **`NAS261nM`** termination messages issued for **PING** console command now display the target IP and DTE addresses
- **`NAS3796I`** messages generated for start of a GATE FC session initiated by an inbound call
- **`NAS3797I`** messages generated for GATE control session activation (bound)
- **`NAS5705W`** messages for RESET SCHEDULED now provides **DIAGX=0000|0001** values
- **`NAS5723W`** messages issued for IDTST= ID String length exception
- **`NAS5724W`** messages issued for IDTST= ‘M’ messages response not received in ‘nn’ seconds
- **`NAS5725W`** messages issued for IDTST= Bad ID condition
- **`NAS5726W`** messages issued for EMSGE option (Datafono) 16 sec PLU response timer expires
- **`NAS7701W`** message now displays optional **RTEIN=CLEAR** forced clear filtering events
- **`NAS7730I`** message showing inbound call request packet data when TRCMCH ICR On set with ALRMSGTXT=SHORT
- **`NAS7731I`** message showing outbound call request packet data when TRCMCH OCR On set with ALRMSGTXT=SHORT
- **`NAS92nns`** messages added for additional authorization processing (please see HNAS Messages and Codes documentation for details)
- **`NAS9205I`** message withheld until today’s date is within 60 days of the EOTDATE
- **`NAS9206I`** message withheld until today’s date is within 60 days of the EOMDATE
- **Clear Request** diagnostic codes **D8 (217)** and **D9 (218)** now contain **DIAGX=** validity codes
- **Clear Request** diagnostic codes **DF (223) DIAGX=8|9** codes added for Datafono

Console, Trace and Debug Enhancements (Commands, Start Parameters...)

- **# text** console input echoed after the HNASCMD-> prefix (example, HNASCMD-> # *text*) comments
- **All** commands now echo their invocation parameters following the HNASCMD-> prefix
- **All** commands now echo their execution parameters following the HNASXEQ-> prefix
- **All** commands that force another command to be executed now use a LIFO rather than FIFO enqueue so that the propagated command is executed immediately after the original command
- **LCLCONS** or ***rconname*** now accepted as the first argument for the **ALARM**, **EXEC**, **SCHEDULE**, **SMSG** and **TRCCONS** commands
- **LCLCONS**, **NASUTIL**, **PING**, **TIMER**, **XOTUTIL**, **XTPUTIL** or ***pcename*** now accepted as the first argument for the **DPCE**, **STATS**, **TRCBFR**, **TRCDATA**, **TRCDISP**, **TRCIO** and **TRCPCE** commands
- **HNASCMD->** command prefix can now be replaced with EXECCMD->, SCHDCMD->, PRXYCMD-> or TRAPCMD-> depending on how the command was scheduled for execution
- **ALARM FILTER=** command now supports (FC) and (FU) message ID suffixes to allow forced conditional (subject to SHOW state) and forced unconditional (SHOW state ignored) SYSCONS display
- **ALARM LOG=?** command now displays date and time of last alert message log entry
- **ALARM LOG=?** now executed automatically during HNAS SHUTDOWN providing alarm log history
- **ALARM MSGTXT={?|LONG|SHORT}** command support has been added to allow the BUILD OPTIONS=ALRMSGTXT= operand to be displayed and/or toggled.
- **ALARM PULSE={?|*|(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)}** command support added to allow dynamic heartbeat message control

- **ALARM LCLCONS|*rconname action*** command support added to allow one console to modify alarm parameters for another console
- **BFR** (internal) trace record now provides buffer allocation and release timestamp and tracestamp.
- **BPM** parm allows all modifiers to be reset before command execution, right side modifiers permitted
- **DLU** command now provides SESSINIT connect status, call set-up direction and improved filtering
- **DLU** command now provides header display information compatible with XOT and XTP resources
- **DLU** command now displays a ‘P’ in place of the leading zero for a PVC VCN value
- **DLU** command now displays the new LUTO column to show the active LU timer in effect (blank implies that no timer is running)
- **DNAS** command now displays additional environment & runtime info, automatic execution at startup
- **DNAS** command now supports **DNAS JCL|DDNAMES** to display the running HNAS JCL
- **DNAS** now executed automatically during HNAS SHUTDOWN providing HNAS information
- **DNAS** now displays the ‘installed under’ system and date stamp in addition to the ‘created under’ and ‘running under’ system and date stamp
- **DNWDF** command added to display active NEWDEFN CDF image maintained in memory
- **DPCE** command now provides header display information compatible with XOT and XTP resources
- **DRMT** command now displays **[NO]CLOSEONAPFAILURE** option for XOT|XTP REMOTEs
- **DRMT** command now displays **[NO]CUD0SELECTSLU** option for MCH|MXT REMOTEs
- **DRMT** command now displays **[NO]DELAYBINDRESP** option for MCH REMOTEs
- **DRMT** command now displays **[NO]IDTST** option for DFX REMOTEs
- **DRMT** command now displays **PVCRECONTMR=**, **PVCSETUPTMR=** and **PVCSETUPREJ=** options for MCH|MXT REMOTEs
- **DRMT** command now displays **[NO]RESETINO** option for MCH REMOTEs
- **DRMT** command now displays **SVCCALLTMR=** option for MCH|MXT REMOTEs
- **DRMT** command now displays **SVC0|5** generic SLU name if one is present for MCH REMOTEs
- **DVC** command now provides SESSINIT connect status, call set-up direction and improved filtering
- **DVC** command now provides header display information compatible with XOT and XTP resources
- **DVC** command now displays a ‘P’ in place of the leading zero for a PVC VCN value
- **DVC** command now displays the new VCTO column to show the active VC timer in effect (blank implies that no timer is running)
- **ECHOXEQ** command added to control how much HNASXEQ> data is echoed back to operator
- **EXEC *ddname*** command allows a list of console command in a HNAS start job ddname to be run
- **EXEC *ddname*** command now stops queuing commands when an embedded EXEC command or the new END command is decoded
- **EXEC *ddname*** command now allows multiple commands to be specified on the same record within the command list file
- **EXEC** (***command1***,***command2***,***command3***,...) allows multiple console commands on a single line
- **EXEC LIST *ddname*** command added to view the content of externally defined command lists
- **EXEC LIST** command has been modified to issue NASC052E error message if ***ddname*** is omitted
- **EXEC STOP** command added to terminate PAUSE command delay mode or CMDLIST execution
- **EXEC *ddname*** command modified to allow CMDLIST comments that start with # to be displayed
- **EXEC LCLCONS|*rconname action*** command support added to allow one console to queue commands and alter queued command processing for another console
- **EXEC** command has been modified to decode the new NOPURGEONERROR and PURGEONERROR keywords
- **HELP** command now displays common command parameters after header message
- **HELP** command new **SHOWALL** keyword displays complete help text for all commands
- **MMEM EOTKEY=*dd...dd*** command allows the <u>TRIAL PERIOD EXPIRATION DATE</u> (**EOTDATE**) for a trial distribution to be extended using the key provided by the ***dd...dd*** digits
- **MMEM EOMKEY=*dd...dd*** command allows the <u>MAINTENANCE/USE ANNIVERSARY DATE</u> (**EOMDATE**) for a permanent distribution to be extended using the key provided by the ***dd...dd*** digits
- **MMEM EOMKEY=*dd...dd*** command also allows a trial distribution to be converted to a permanent distribution dynamically using the key provided by the ***dd...dd*** digits

- **MON TAP** command NAS251*n*M monitor messages can now be filtered using the ALRMFLTR= operand so that they can be optionally routed to SYSCONS using the (FC) or (FU) suffixes
- **MON TAP** command now accepts **PKTDATA|MAXDATA|MINDATA|NODATA** as arguments to control the type of information displayed in the NAS251*n*M monitor messages
- **MRMT LOGTAB=*lgtbname*-R** command now permits the named LOGON table to be reloaded into memory
- **MRMT LUNAME=|SVC0=|SVC4=|SVC5=*sluname*-{A|I}** command now permits manipulation of SLU state (A is for active (online) and I is for idle (offline)) **-** VARY command is executed
- **MRMT LUNAME=*** command now permits manipulation of asterisk (*****) for GATE (LLC4) support
- **MRMT *rmtname* IPADDR=*aaa.bbb.ccc.ddd*** dynamic TYPE=XOT REMOTE IP address changes
- **MRMT OPTIONS=[NO]CLOSEONTAPFAILURE** command allows a customer to toggle this option
- **MRMT OPTIONS=[NO]CUD0SELECTSLU** command allows a customer to toggle this option
- **MRMT OPTIONS=[NO]DELAYBINDRESP** command allows a customer to toggle this option
- **MRMT OPTIONS=[NO]IDTST** command allows a customer to toggle this option
- **MRMT OPTIONS=NRITAB=*nrtbname*-R** command now permits the named NRI table to be reloaded into memory
- **MRMT OPTIONS=PVCRECONTMR=*secs***, **PVCSETUPTMR=*secs*** and **PVCSETUPREJ=*code*** allows a customer to delete, change or add these options for TYPE=MCH|MXT REMOTEs
- **MRMT OPTIONS=[NO]RESETINO** command allows a customer to toggle this option
- **MRMT OPTIONS=SVCCALLTMR=*secs*** allows a customer to delete, change or add this option
- **MRMT SVC0|5={(*sluname*+*gluname*/...)}** command now allows generic SLU name to be added for TYPE=MCH REMOTE
- **MRMT USSTAB=*ustbname*-R** command now permits the named USS table to be reloaded into memory
- New **PARM='DBUG TCP'** start parameter allows certain TCPIP 198 ABENDs to be bypassed, replaced by new NAS2109S and NAS2110S alarm messages - HNAS execution continues
- **PAUSE *seconds*** command suspends execution of console command following PAUSE command
- **PING** command now produces NAS261*n*M monitor messages that can be filtered using the ALRMFLTR= operand so that they can be optionally routed to SYSCONS using the (FC) or (FU) suffixes
- **PING (XOT) *dmyname*** allows predefined pinging xot call request values to be employed (all fields)
- **PRNT** command **PRTSWLST=(?|*|list)** displays, clears, switches, updates SYSPRINT controls
- **PRNT** command now support the **NEXTPRSW/SWITCHNOW** to force a switch to next ddname
- **PRNT** command now support the **RSMEPRSW** parameter to restart SYSPRINT switching
- **PRNT** command **TIMESTAMP=(*time-format*,+D|-D)** updates TIMESTAMP time and date controls
- **PRNT ON|OFF** command now forces NAS021*ni* alarm message to be issued
- **PRNT DATE ON|OFF, QLLC ON|OFF ...** command now forces NAS0230W alarm message to be issued
- **QUIT - QE [*ddname*|(*cmd1*,*cmd2*...*cmdn*)** command executes command list before shutdown
- **QUIT** command now unconditionally executes **DNAS** and '**ALARM LOG=?**' as part of the shutdown process
- **SCHEDULE {*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)|LIST|SUSP|RSME|PRG}** console command support added to allow scheduled console commands to be added, deleted or displayed
- **SCHEDULE LCLCONS|*rconname action*** command support added to allow one console to schedule commands and alter scheduled command processing for another console
- **SHOW CMSG ON|OFF** command added to manipulate the **SHOWCMSG** start parameter
- **SHOW ON|OFF|ERR** command now forces NAS011*ni* alarm message to be issued
- **SHOW MORE|LESS** command now forces NAS0130W alarm message to be issued
- **SHOWCMSG** parameter allows configuration and alarm message multiple blanks compression
- **SMSG LCLCONS|*rconname* '*text*'** command support added to allow the target console to be identified by PCE name
- **SNAP ALL|*trgtlist*** command support added to allow specific HNAS storage areas to be dumped in the SYSPRINT file
- **TCPIP** external interrupt table entries extended to allow logging of additional diagnostic information making HNAS TCPIP tracing less of a requirement for problem diagnosis

- **TRCCONS LCLCONS|*rconname* {ON|OFF}** command support added to allow the target console to be identified by PCE name
- **TRCLU**, **TRCMCH**, **TRCMCHX** & **TRCVC** provide **HNASXEQ->** modifiers and arguments in effect
- **TRCMCH ICR OCR ICLR OCLR** now supported on **PARM=** Start Parameter
- **TRCMCH ICR OCR ICLR OCLR** now have *local* or *global* significance (Default|LCL|GBL argument)
- **TRCSUBR *eventlist*** now supported on **PARM=** Start Parameter and as console command
- **TRCTRAP ALRMLIST=(NAS*nnnni*/*data*/*data-offset*,...)** command, refined trap message filtering
- **TRCTRAP ALRMLIST=(NAS*nnnni*/'*data*'/*data-offset*,...)** command (***data*** as quoted string)
- **TRCTRAP TRAPACTION=(EXEC=*ddname*)** option support command list execution upon trap
- **TRC*type*** console commands (i.e. TRCMCH ICR OCR ...) now accept multiple arguments
- **VARY *mchname* {OFF|ON}** command allows MCHs to be disabled or enabled
- **VARY *sluname* {OFF|ON}** command permits or restricts SLUs for inbound or outbound calls
- **VARY *sluname* FORCE** command has been modified to close the TCPIP socket associated with the SLU even if the SLU itself is inactive (normally only PVCs can be in this state)

HNAS V2R3M0 Release Summary:

Functional Enhancements

- SMP/E Product Installation Supporting HNAS private global zone
  Common global zone support was added on 03-31-2004.
- Extended Diagnostic Reason Code Support (Improved session Call and Clear Troubleshooting)
- CART= WTO Support for Session Manager (Netview, TDSLink) Operator Consoles (ALRMCART=)
- Improved Console and SYSPRINT output filtering using new PRNT *type* and SHOW CONS settings
- SYSPRINT Dataset Recording Improvements
- SYSPRINT Log Filtering
- Start parameter follower support
- XOT TAP= support now provides calling/called address, facilities and call user data parameters
- TRCTRAP= support provides automation tools for improved debugging and trace suspension
- NETVIEW routing support, asynchronous alarm messages/synchronous console command output
- TRCTRAP= enhancement provides optional SNAP dynamic dump of control blocks for trap cond.
- NETVIEW routing enhancement for asynchronous alarm messages/synchronous console command output that allows WTO routing code to be configured (OPTIONS=WTOROUTCDE(*type*)=*code* on BUILD, where *type*=ALRM|CONS and *code*=0-128)

Configuration Enhancements

- Callout Session Connect Balancing Extension (OPTIONS=BALANCERTEOUT)
- Backup Router Support for LLC0/5 Callout
- CUD=, FAC=, DCEADDR= and DTEADDR= TAP parameters added for TYPE=XOT REMOTEs
- OPTIONS=INHIBITBIDREJ - optional BID unconditional accept logic
- OPTIONS=NORTRBIDREJ - optional BID 0814 sense reject logic
- OPTIONS=ONEPIUINB - option to force one PIU per Bracket
- OPTIONS=REQSESSDELAY=*value* - option for request session delay timer value
- OPTIONS=RESIDSTART=*decimal-start-value* - option to assign start values for GATE Resource ID
- INIT=({ACTIVE|IDLE},DELAYTIME=*minutes*,RETRYLMT=*count*) added for LOCAL definitions
- HNAS CDF Configuration Error Summary Improvements
- HNAS CDF Configuration Error Processing, NAS1999W|E|S improves error message identification
- FASTRUN VTAM operand generation enhancement for QLLC TYPE=SPU REMOTE resources
- FASTRUN AMNF VBUILD statement name from BUILD APPLNAME= rather than NASNAME=
- IDLETO=*minutes* added for non TYPE=XOT REMOTE level definition statements
- The configuration process will now accept a pound sign (#) in cc-1 improving CDF comments
- Start Parameter PFXWTO CONS causes console output to be prefixed with the NASNAME=value
- Start Parameter PFXWTO *text* allows user defined alert message and console output prefix name
- OPTIONS=PFXDCEADDR appends DCEADDR= in front of GATE outbound *calling* DTE address
- OPTIONS=LLC0CTCPCHK/LLC5CTCPCHK DATE like control LLC0|5 Call Accept CTCP sessions
- CONLMT=0 can now be coded on BUILD to disable Remote Console access
- Bidirectional **T**woway (Callin/Callout) VC support for PCNE/PAD (SVC0=/SVC5=) resources
- XOT PVC= support now includes mxtname assignment for user defined window/packet size values
- OPTIONS=REUSEBSYSPU supports special QLLC call-to-call (already connected VC) condition
- SVC0=, SVC4= and SVC5= parms now allow SLUname generation using prefix and suffix values
- TRAN= parm now accepts NPSIEVEN, NPSIODD, NPSIMARK, NPSISPACE for NPSI tables
- RTEIN= now supports calling (=>S) source address in addition to the existing (=>T) target address
- OPTIONS=BALANCERTEIN supports LLC-n RTEIN= MCH level round robin session balancing

Alert/Alarm Message and Clear/Reset Code Enhancements

- Message Filtering enhancement for NAS***innns*** messages
- Julian Date (*nnn*) Now Available in Alert Message Entries

- NAS2021W SERVER INITIALIZATION FAILED, LOCAL VARIED OFFLINE message added
- NAS2601E SOCKET POOL DEPLETED/NAS2602I SOCKET POOL RESTORED messages added
- NAS8000I Starting Session message modified to improve QLLC call setup status notification
- NAS7718T, NAS7719T and NAS7798T Inbound/Outbound call request packet content trace messages via TRCMCH ICR|OCR
- NAS0050A, NAS0060W and NAS0070W alert messages provided for TRCTRAP support
- NAS3703W NOTIFY and NAS3704W CLEANUP (REQSESS related) messages added
- NAS7715W Call Request Failed message now contains extended diagnostic event reason codes
- NAS7713W SECOND CLEAR FROM ip-addr(port)... message added improving event reporting
- Clear Diagnostic codes
  Added diagnostic code 221 x'DD' - Timeout. CTCP did not UNBIND after receiving clear
  Added diagnostic code 222 x'DE' - Second clear received from router for VC
- NAS7797W CALL FROM... XOT TCP/IP message added improving event reporting
- NAS3797I LU sluname RECEIVED BIND FROM PLU pluname PVC messages added
- NAS4706W LU sluname REJECTING BIND FROM PLU pluname SENSE=xxxx message added
- NAS0001W PARMLIST OMITTED message added indicating PARM= values omitted
- Clear diagnostic code 211 x'D3' logic added to address non recoverable packet Reset conditions
- NAS3705W LU lu-name REJECTING message added indicating that PIU is being rejected
- NAS7774W PVCSETUP FAILED message when remote issues socket close in response to setup
- NAS7718T, NAS7719T and NAS7798T Inbound/Outbound pvc setup content trace messages
- HNAS PVC Setup Status Codes were reassigned improving router setup retry processing
- NAS7795T Inbound/Outbound clear request packet trace messages via TRCMCH ICLR|OCLR
- NAS251xM TAPping (Keep Alive) monitor messages via MONTAP/MON TAP
- NAS4710W lu-nm LU st-addr SENDING DIAG PKT :text BFR NEXT alert message added
- NAS3705I LU lu-name REJECTING cmd #seq SENSE=bbbbbbbb BID alert message added
- NAS4707W sluname GENERATING ERR/INFO PACKET FOR CTCP alert message added
- NAS4708W GATE FC CTL SES LU luname CLEARED BY CTCP alert message added
- NAS4709W REMOTE LU LUIQ TIMEOUT,LUIQ BFR alert message added
- Clear diagnostic code 212 x'D4' logic added to report VC session already active conditions
- NAS3797I messages generated for GATE control session activation (bound)

Console and Trace Enhancements

- Error Reporting Enhancement
- Resource Name follower supersedes RNM=, LNM= or LUNM= Console Command Modifiers
- DMAP APAR command executed at startup, now generates APAR ID summary output
- DNAS command now displays APAR through information as well as sorted APAR list
- TRCLU command improvements for TYPE=SPU resources
- DLU command now displays SLU association with TYPE=SPU REMOTE definitions
- DPCE command now displays PCE association with TYPE=SPU REMOTE definitions
- DPCE, some TRC*types* and STATS commands now accept various arguments relative to PCE
- DLU, DVC and DPCE commands now accept TRACE=YES|NO arguments
- New PING XOT Command Allows Addressing of Routers IP Address, Protocol and MCH
- MRMT Expanded REMOTE LUNAME= Config. Parameter/Resource Reassignment (QLLC LLC3)
- MLCL & MRMT commands have been modified to permit updating of the INIT= operand
- VARY command has been modified to allow a LOCAL to be varied offline or online like REMOTEs
- DTRC command now displays the trace table wrap count, WRAPCNT=*wrap-cnt* in the trace header
- DLU, DMCH, DPCE and DVC commands now accept SHOWtype arguments for improved status
- TRCLU trace entries now provide keywords (BIND, SDT, UNBIND) to improve search capabilities
- DMCH command now accepts FMT3 as an argument providing MCH and LU association displays
- PFXWTO *name* option allows user defined alarm messages & console output prefix name
- TRCMCH ICR and OCR trace option now produces formatted call request packet message content
- ALRMFLTR console command now appends instead of replacing messages from list array
- MLCL *lcl-name* RTEIN= and RTEOUT= operands now permits insertion/deletion of table entries
- DNAS command now produces a list of missing *apar-ids* as well as individually specified *apar-ids*

- New DNAS USERMODS command produces a list custom-user-mods on system, as appropriate
- DNAS command now provides HNASOBJX and HNASOBJX distribution USERMOD indicator
- MRMT expanded allowing PCNE/GATE/PAD (SVC0|4|5=) SLU names to be added/chgd/deleted
- SHOW MORE/SHOWMORE console/start parameter provides expanded TCP/IP event debugging
- TRCALL command modified to remove PCE trace function for option ON|OFF, reducing confusion
- DNAS command now provides distribution type (SMP/E or NON-SMP) on the header line
- VARY command has been modified to allow specific sockets or ranges to be varied offline or online
- TRCMCH ICR and OCR trace option now produces formatted PVC setup packet message content
- DVC command now displays PVC setup packet exchange status
- MRMT command now allows TRAN= parm to be updated 'on the fly' (all TRAN= values accepted)
- TRCMCH ICLR and OCLR trace option now produces NAS7795T messages for all clear packets
- MON TAP ON|OFF|ALLON|ALLOFF command produces NAS251xM KeepAlive monitor messages.
- VARY command now allows closure of active sockets (LOCAL|REMOTE) with new FORCE option
- DPARM command now accepts MODIFIERS as an argument to restrict display to modifiers only
- HELP command text localized into respective help command modules improving maintenance
- DPARM command now displays ALLON|ALLOFF instead of ON|OFF for some TRCtyp states
- PRNT QLLC console command start parameter PRNTQLLC added to filter NAS8xxxx messages
- DNAS command now provides Host OS & Version information and Execution date in display output

<u>QLLC Enhancements</u>

- QLLC Callout via TYPE=SPU REMOTE
- QLLC Callout via TYPE=MCH REMOTE
- QLLC Callout Support - Calls initiated via Application Bind to designated Control LU
- QLLC Callout Support - Calls initiated via various Timer Control Options
- QLLC Callout Support - Calls initiated via HNAS Console Command Controls
- QLLC Callout Support - Calls initiated when HNAS is Started/Activated
- QLLC TYPE=SPU LUNAME= SLUname MXTname association (wire an SLU to an application)
  (LUNAME= operand enhancement for QLLC SPUs (SLU *applid* and *mxtname* support)
- QLLC TYPE=SPU REMOTE IDLETO=*minutes* Inactivity Clear Support
- QLLC TYPE=SPU LUNAME= coding improvement for LOCADDR values (gap-count vs. commas)
- QLLC TYPE=SPU APPLNAME=ACQUIRE Printer Bind Support Improvement
- QLLC TYPE=SPU SVC3=ALLOW option improves MCH SPU coding and operation

HNAS V2R2M0 Release Summary:

- REGION Size and CPU Utilization Improvements
- High Memory Support
- Multiple 'same type' Server Support
- Multiple Stack Support
- Shared Socket Support
- Authorization Key Support (trial users distribution)
- XOT QLLC PU Type 1|2, LU Type 0|1|2|3 device Support (PUT2.1 or LU 6.2 not available)
- FASTRUN AMNF Generation
- Multiple SYSOUT Support
- Callout Session Connect Balancing
- Callout Alternate DTE Connect Retry Support
- Callout Connection Routing using Calling or Called Addresses
- Callin DTE Address Filtering Support
- Callin SLU/PLU Fixed Connection Support
- Callin Default PLU Assignment Support (SYSL= now optional)

- Gate Control Session parameter to fine tune the Activation Delay Timer
- Global Buffer List Table Size parameter to reduce memory requirements
- Extensive Console Alert Filtering/Reduction
- Alert Message Additions and Enhancements
- Improved Diagnostic capabilities, Improved Options and various other enhancements
- TAP= XOT protocol level tapping improvements
- SUBD= allowed on non-Fast Connect GATE MCH to permit CTCP selection by subaddress digits
- FASTRUN AMNF VBUILD statement name now comes from APPLNAME= rather than NASNAME= operand on BUILD definition statement (APAR 2200080).

HNAS V2R1M1 Release Summary:

- Maintenance Update
- Z/OS V1R2 (and V1R4) Support
- SLU Selection by CUD Data
- SLU to PLU Connection Wiring
- MCHTMR Support
- Minor Options/Enhancements

HNAS V2R1M0 Release Summary:

- Dynamic IP Address Assignment (Call-in) Support
- LOGTAB Support
- Performance Improvements
- Remote Console Alarm
- Monitoring, Local Console Alert Filtering
- Improved Options and various enhancements

HNAS V1R1M4 Release Summary:

Please refer to the 114 documentation manual for the list of new features.

Please refer to the specific ‘HNAS VnRnMn - New Features Overview’ for a brief description of all enhancements.

HNAS V2R4M0 New Features

V2R4M0 General Availability - 2006-07-31

Functional Enhancements:

- All appropriate V2R3M0 **APARs** have been incorporated as permanent fixes in HNAS V2R4M0 as of the general availability date. The HNAS product is continually updated using Standard APARs (bug fixes) and Enhancement APARs (enhancements) in an effort to provide a stable product as well as product improvements and new features for our customers. The use of enhancement APARs is a way to provide these product updates without having to wait for a new VnRnMn release (Note that some customer enhancement requests are deferred to a later release because the changes are either too big or inappropriate as an Enhancement APAR).

- **Non-SMP/E installation and maintenance improvements**

  The process that we employ to generate non-SMP/E HNAS 240 edistributions has been improved. A REXX exec is provided which, when customized, generates the jobs required to allocate data sets and to install and maintain HNAS. Non-SMP/E and SMP/E product edistributions are now built using a common interface providing a uniform installation process.

- **SMP/E fully integrated in the installation and maintenance documentation**

  The SMP/E installation and maintenance instructions are now fully integrated into the standard HNAS documentation series (including content improvements in the maintenance section).

- **ISARX25 Datafono Support Integrated into the Standard HNAS Product (IBM SPAIN)**

  This support was previously available as a custom 230.c add-on to the HNAS 230 product. This support was developed for customers in Spain who are using the Datafono ISARX25 implementation and required a HNAS solution to migrate from their 3745 communication controllers. The Datafono option must be ordered prior to shipment of HNAS because a special distribution key is required to enable the support. The DNAS display output contains the text **`'DATAFONO SUPPORT IS INCLUDED'`** if the support is enabled.

  The Datafono documentation and installation instructions are now included in the standard HNAS documentation series (in 230.c these instruction were in a separate add-on booklet).

  With APAR 2400011 installed Datafono customers can easily install custom versions o f the XAICDTFT translate table CSECT (see XAICDTFT in the index)

  With APAR 2400074 the IDTST and CUD0SELECTSLU options are supported. These options provide features (message ID checking and CUD0 participation in LU selection) provided by ISARDX25.

- **BUILD CONCMDQ=*ddname* (CMDLIST)** support permits lists of console commands to be precoded in JCL DD statements outside of the CDF file. While the command list support is primarily utilized via the new **EXEC** console command there are other console processes that utilize the command lists. Following is an overview of the new CMDLIST features while a detailed description can be located under the Configuration Enhancements and Console and Trace Enhancement headings in this section:

- Many (but not all) start parameters have a console command equivalent. At start-up, you have the option of specifying console command-like start parameters in the PARM= operand (or in the file identified by the PARMFILE= parameter - see below) or by specifying the equivalent console command in the BUILD EXEC= (or CONCMDQ=) operand. Prior to the introduction of the PARMFILE= parameter, the 100 character JCL limit imposed by z/OS on the EXEC PARM= operand made it necessary to use the BUILD EXEC= operand in conjunction with the EXEC PARM= operand so that the 100 character limit would not be exceeded. Now that you can specify all start parameters in a parameter file, the need for listing parameter equivalent console commands in the BUILD EXEC= operand becomes less necessary although still equally valid.

- A list of commands can be executed via the operator console by entering **EXEC *ddname***. This command can also be provided at startup via CONCMDQ=('EXEC *ddname*'). Equivalently, you can also specify CONCMDQ=*ddname*. The difference is that the *ddname* identified in the CONCMDQ=('EXEC *ddname')* is not processed until the EXEC command is executed while the *ddname* identified in CONCMDQ=*ddname* is processed when the CDF is scanned.

- A list of commands can be executed via the new **TRCTRAP TRAPACTION=(EXEC=*ddname*)** option.

- **QUIT** command now accepts **E**xec as a new keyword follower. **QE {*ddname*|(*cmd*1,...,*cmd*n)}** causes command lists to be executed before HNAS is shutdown.

Note: Currently **BUILD CONCMDQ=(EXEC *ddname*)** and command line **'EXEC *ddname*'** are restricted to a single 'EXEC *ddname*' reference. CONCMDQ=(EXEC *ddname1*,EXEC *ddname2*,...) or a command line of 'EXEC *ddname1* EXEC *ddname2* ...' is not supported as a concatenation. When multiple EXEC *ddname* statements are decoded in a single list, all but the first are ignored. If you want command lists to be concatenated, you can either concatenate the command lists in JCL, merge the command list files or include the next command list as the last command in the current command list. For example, you could end the command list identified by *ddname1* with EXEC *ddname2*. We plan to expand this support in a future 240 Enhancement APAR by allowing multiple EXEC queues. This will permit multiple lists to be concatenated via the CONCMDQ= operand or via command line input. CONCMDQ=(EXEC *ddname1*,EXEC *ddname2*,...) and command line input of 'EXEC *ddname1* EXEC *ddname2* ...' will operate as the operator expected.

- **PARM='EOTKEY=*dd*...*dd*'** allows the TRIAL PERIOD EXPIRATION DATE (**EOTDATE**) for a trial distribution to be extended using a special 16 decimal digit key provided by the ***dd*...*dd*** digits. An HNAS trial distribution 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 refresh distribution. The new EOTKEY= parameter now allows the EOTDATE to be extended without the need of a new refresh distribution. The EOTKEY=*dd*...*dd* digits are provided by Comm-Pro in a special file that is sent as an e-mail attachment or is downloaded from our FTP server. The EOTKEY=*dd*...*dd* string can then be cut and pasted to the PARM= operand (or MMEM console command - see page D-43). 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
```

The DNAS display for a trial distribution has the following form (for example):

```
HOST NAS INFORMATION FOLLOWS
     HNAS VERSION=V2R4M0 DIST=SMP/E                                  1
     HNAS PROGRAM RUNNING UNDER z/OS 01.11.00                        2
     HNAS PRODUCT INSTALLED UNDER z/OS 01.11.00                      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 2010/11/29                  8
     HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 2400106      9
     MOST RECENT MAINTENANCE APPLIED IS APAR 2400106                10
     AUTH=032D SHIPID=1100000011199999                              11
     CUSTID=SFD_99999                                               12
     CUSTINFO=COMM-PRO ASSOCIATES                                   13
     TRIAL PERIOD EXPIRATION DATE IS 2010/12/31                     14
     DATAFONO SUPPORT IS INCLUDED                                   15
                                                                    16
                                                                    17
     APARID   MAINTENANCE STATUS                                    18
     ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400106            19
```

If EOTKEY=4961000737880526 (for example) is used to extend the EOTDATE, DNAS display lines 11, 14 and 16 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                           16
```

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

- **PARM='EOMKEY=*dd...dd*'** allows the <u>MAINTENANCE/USE ANNIVERSARY DATE</u> (**EOMDATE**) for a permanent distribution to be extended using a special 16 decimal digit key provided by the ***dd...dd*** digits. An HNAS permanent distribution 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 EOMKEY= parameter now allows the EOMDATE to be extended without the need of a new refresh distribution. The EOMKEY=*dd*...*dd* digits are provided by Comm-Pro in a special file that is sent as an e-mail attachment or is downloaded from our FTP server. The EOMKEY=*dd*...*dd* string can then be cut and pasted to the PARM= operand (or MMEM console command - see page D-43). The EOMKEY file has the following format:

```
EOMKEY=4962030747980516
HNAS EOMKEY CREATED AT 16:00:19 ON 2010/11/28
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31
CUSTID=SFD_99999
CUSTINFO=COMM-PRO ASSOCIATES
EMKYDC=0201012311199999
```

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

- **PARM='EOMKEY=*dd...dd*'** also allows a trial distribution to be converted 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 16 above as follows:

```
AUTH=000  SHIPID=1100000011199999 EMKYID=1100000011199999      11
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/12/31*                14
EOMKEY=4962030747980516 IS IN EFFECT                           16
```

  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.

-
- **PARM='PARMFILE=*pfddname*'** allows all or some HNAS start parameters to be provided in a sequential file. This means that you can specify all HNAS start parameters without regard to the 100 character limit imposed by the JCL scanner for the EXEC PARM= operand. Please refer to Chapter 2 of the HNAS Guide and Reference manual for more information.

  Note: PARMFILE= parameter logic was introduced into 240 via APAR 2400094.

- **BUILD PRTSWLST=(*controls*)** are now provided in HNAS permitting automatic SYSPRINT switching when the current SYSPRINT log file becomes full, when the designated time occurs or when forced by the console operator. Please refer to the PRTSWLST= entry provided in Configuration Enhancements section for additional information.

- **REMOTE TYPE=DMY** was added to permit future expansion of some console command parameters. PING (XOT) dmyname is the first command to use this remote type.

Note: Please refer to the sections (Configuration Enhancements, Alert/Alarm Message and Clear/Reset Code Enhancements, Console and Trace Enhancements and Enhancements - Expanded Content) in this 240 New Features document for additional information regarding these and other new or enhanced features.

Configuration Enhancements:

Various **CDF Options** and **Parameters** were added or changed to allow the user greater flexibility when defining resources in V2R4M0:

- **ALRMFLTR=** table has been expanded from 16 to 128 entries allowing fine tuning of alert messages filtering for SYSCONS and SYSPRINT message routing or purging.

- **ALRMFLTR=** now supports (FC) and (FU) message ID suffixes which allow alarm messages to be forced conditionally to SYSCONS (subject to SHOW state) or forced unconditionally to SYSCONS (SHOW state ignored). For example, if ALRMFLTR=(...,NAS2020I(FC),NAS0299I(FU),...) is specified, NAS2020I alarm messages will be sent to SYSCONS except when SHOWOFF is effect while NAS0299I alarm messages will be sent to SYSCONS regardless of the SHOW state.

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

- **CONCMDQ=** can now include the new **EXEC *ddname*** command which permits a list of console commands to be pre-coded in JCL DD statements outside of the CDF file. Please refer to the **EXEC *ddname*** command for additional information.

- **CONCMDQ=*ddname*** can now be specified to provide a dataset command list (provides consistency with the SCHEDULE=*ddname* operand).

  Syntax: CONSMDQ={*ddname*|(*cmd*,...,*cmd*)}

  (*cmd*,...,,*cmd*) is an inline command list (no change).

  ddname identifies a file that contains a list of commands. A *ddname* command list can contain the following records:

  Comments are allowed and start with an asterisk (*), semi-colon (;) or pound sign (#) in record column 1. Comments can also appear on a command line but must start with a semi-colon after the command. A single command can be specified on a single record as follows:

  cmd ; *comment*

  Multiple commands can be specified on a single record as follows:

  (*cmd* , ... , *cmd*) ; *comment*

  Note: The new CONCMDQ=*ddname* operand support was introduced into 240 as Enhancement APAR 2400080.

- **EXEC=** operand has been added as an alias for CONCMDQ= (provides name symmetry with the EXEC command like the SCHEDULE= operand and SCHEDULE command).

  Note: The new EXEC= operand support was introduced into 240 as Enhancement APAR 2400080.

- **EXEC=** operand has been modified to decode the new NOPURGEONERROR and PURGEONERROR keywords. One of these keywords may be entered as the first value in an inline command list (**NOPURGEONERROR|PURGEONERR,*cmd*1,...,*cmd*n**) or as the first non-comment record in a *ddname* command list.

  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 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
```

5) 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.

- **FASTRUN CONCMDQ** support provides the ability to execute a string of console commands (CONCMDQ=*cmdvalues)* during the FASTRUN process. This can be very useful to set, display and test various console commands sequences after the CDF scan without actually starting HNAS (HNAS doesn't initiate any conversations with VTAM resources or TCPIP Stack processes in FASTRUN mode).

- **FASTRUN AMNF** generation support now propagates VTAM parameters from the MCH to all GATE control session SLU APPL statements for the MCH. Prior to this change, the VTAM parameters for GATE control session SLUs defaulted to DLOGMOD=INTERACT,EAS=1 only.

  Note: GATE control session SLU VTAM parameter change was introduced into 240 as Enhancement APAR 2400029.

- **FASTRUN AMNF** generation support now propagates VTAM parameters from the MXT associated with a PVC SLU to the SLU APPL statement (MCH VTAM parameters used when MXT not coded). Prior to this change, the VTAM parameters for PVC SLUs came from the MCH only.

  Note: PVC SLU VTAM parameter change was introduced into 240 as Enhancement APAR 2400029.

- **INIT={ACTIVE|IDLE}** is now valid for TYPE=MCH REMOTE definition statements. INIT=ACTIVE is the default. INIT=IDLE will cause the MCH to be initially inactive. No calls to or from this MCH can take place until it is activated via the new VARY MCH ON console command (see below) or the MRMT INIT=ACTIVE console command which invokes VARY MCH ON.

- **INIT={ONLINE|OFFLINE}** is now allowed as alternate values for ACTIVE|IDLE, respectively. The **DPCE** console command display for a LOCAL or REMOTE in IDLE state will depict OFLN as the current socket state.

Note: INIT=ONLINE|OFFLINE support was introduced into 240 as Enhancement APAR 2400014.

- **LUNAME=(...,*sluname*{-{A|I}},...)** is now allowed to specify an initial state for the named SLU.

  -A (or omitted) is for active and -I is for inactive or idle with active being the default.

  Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

- **LUNAME=** operand has been modified to accept a **GENTYPE** value which is used to control hex versus decimal SLU name generation: LUNAME=(*sluname*-{A|I}/*pluname*/*pfxlu***-{H|D**}/*sfxst*/*cnt,*...)

  When *pfxlu*-D is specified, SLU names are generated in ascending decimal order starting with the *sfxst* value for number of SLUs identified by the *cnt* value.

  When *pfxlu*-H or *pfxst* (without the -H) is specified, SLU names are generated in ascending hexadecimal order starting with the *sfxst* value for number of SLUs identified by the *cnt* value. This is how SLU names have always been generated prior to APAR 2400035.

  Note: *pfxlu*-{H|D} support was introduced into 240 as Enhancement APAR 2400035.

- **OPTIONS=ALRMSGTXT={LONG|SHORT}** has been added to the BUILD definition statement. This OPTION allows you to select the long (default) or short 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. The new OPTIONS=ALRMSGTXT= operand effects various error and informational alarms.

  For callin/callout informational messages (**NAS7*nnn*I**), see their description later in this chapter.

  For TCPIP (**NAS2*nnns***), VC (**NAS5*nnns***) and QLLC (**NAS8*nnns***) alarm messages that have the following LONG format:

  ```
  NAScnnns comp=iii.iii.iii.iii(port) SOCKID=sockid PCEID=pceid NAME=compname
  NAScnnns information text
  ```

  The new SHORT format will appear as follows:

  ```
  NAScnnns comp=iii.iii.iii.iii(port) NAME=compname information text
  ```

  As you can see, the SOCKID= and PCEID= fields have been removed for the SHORT format so the alarm message can fit on a single line.

  For these messages, ***c*** is the component ID (2|5|8), ***nnn*** is the message number, ***s*** is the message severity (I|W|E|S) and ***comp*** is the component type (SERVER or CLIENT).

  Note: BUILD OPTIONS=ALRMSGTXT={SHORT|LONG} support was introduced into 240 as Enhancement APAR 2400068.

- **OPTIONS=DELAYBINDRESP** has been added to the TYPE=MCH REMOTE statement. This OPTION causes HNAS to delay the response to a BIND that triggers an LLC0 or LLC5 callout operation until the call succeeds (call accept received, +RSP) or fails (clear or is timeout, -RSP). The sense data for a -RSP will be 0801C3D9 - resource not available.

- **OPTIONS=CUD0SELECTSLU** has been added to the TYPE=MCH REMOTE definition statement. This OPTION allows the CUD0 byte from an incoming Call Request packet to participate in LU name selection. This provides a capability found in ISARDX25 (IBM Spain) 37xx support.

  Note: OPTIONS=CUD0SELECTSLU support was introduced into 240 as Enhancement APAR 2400074.

- **OPTIONS=IDTST** has been added to the TYPE=DFX REMOTE definition statement. This OPTION implements the ISARX25 Datafono IDTST option which ensures that PLU messages are being sent to the correct SLU.

  Note: OPTIONS=IDTST support was introduced into 240 as Enhancement APAR 2400074.

- **OPTIONS=NOCLOSEONTAPFAILURE** has been added to the TYPE=XOT|XTP REMOTE definition statement. This OPTION allows active sockets to remain active when router contact is lost after 2 consecutive TAP (Keep Alive) failures. This option was added for customers who would like to activate TAP (XOT Keep Alive) processing without taking a REMOTE down (out of service) if a TAP Contact Lost failure occurs.

  Note: OPTIONS=NOCLOSEONTAPFAILURE support was introduced into 240 as Enhancement APAR 2400055.

- **OPTIONS=PVCRECONTMR=*secs***, **PVCSETUPTMR=*secs*** and **PVCSETUPREJ=*code*** have been added to the TYPE=MCH|MXT REMOTE definition statement.

  PVCRECONTMR= specifies how often HNAS tries to re-establish a session between a PVC VC and the PLU. The VTAM session with the PLU is established after the PVC session between HNAS and the router is established by PCV SETUP packets. If the PVC's VTAM session is ended (PLU UNBIND, Notify or TPEND) then the value specified by this operand controls how often HNAS will try to reconnect the session with the PLU. If PVCRECONTMR= is omitted for a TYPE=MCH REMOTE, a default value of 60 seconds will be used. If PVCRECONTMR= is omitted for a TYPE=MXT REMOTE that is associated with a PVC, no default value is set so that the root MCH value will be used.

  PVCSETUPTMR= specifies how often HNAS should send PVC SETUP packets to establish a PVC session between HNAS and the router. The option only has meaning when the PVC definition in the HNAS CDF identifies the router to be used for the session (see PVC= operand). If PVC-SETUPTMR= is omitted for a TYPE=MCH REMOTE, a default value of 60 seconds will be used. If PVCSETUPTMR= is omitted for a TYPE=MXT REMOTE that is associated with a PVC, no default value is set so that the root MCH value will be used.

  PVCSETUPREJ= specifies a PVC SETUP reject status code that will be used any time HNAS receives a SETUP for the PVC. Since a status code greater than X'0F' causes the router to stop sending SETUP packets for the PVC, this option may be used to shut down PVC SETUP initiation by the router. If PVCSETUPREJ= is omitted, SETUP packets are processed normally (status code to connect or reject the session generated by HNAS). This parameter was initially implemented for PVC setup testing but also includes benefits as described above.

  Note: OPTIONS=PVCRECONTMR=,PVCSETUPTMR=,PVCSETUPREJ= support was introduced into 240 as Enhancement APAR 2400059.

- **OPTIONS=RESETINO** has been added to the TYPE=MCH REMOTE definition statement. This OPTION specifies that HNAS is to terminate the session with the PLU when a RESET is received from the remote.

Note: OPTIONS=RESETINO support was introduced into 240 as Enhancement APAR 2400081.

- **OPTIONS=SVCCALLTMR=*secs*** has been added to the TYPE=MCH|MXT REMOTE definition statement to specify how long HNAS should wait for a response to a transmitted Call Request packet before it assumes that the call has failed because there was no response. Note that T21= can be specified as an alternate name for the SVCCALLTMR= suboperand.

  Note: OPTIONS=SVCCALLTMR= support was introduced into 240 as Enhancement APAR 2400069. Prior to the APAR a fixed 30 second timer was used.

- **OPTIONS=(TIMESTAMP=*format*-{+D|-D}** was added to the BUILD definition statement to enhance SYSPRINT timestamp resolution.

  Note: Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

- **PARM=** (**EXEC**) operand now supports abbreviated start parameters reducing the likelihood that the **PARM=** operands 100 character limit will be exceeded. For example, **TLU MXDT** can now be specified instead of **TRCLU MAXDATA** which conserves 5 characters of PARM= operand space.

  Note: Abbreviated PARM= operand support was introduced into 240 via APAR 2400048.

- **PARSE** definition statement has been added to allow customers to specify their own parsing characters that will override established HNAS parsing characters for the CDF scan. The PARSE definition statement must be the first definition statement in the CDF and must precede any CDF records that utilize the special parsing characters that it defines. After the PARSE definition statement is processed, the override parsing characters will go into effect turning the established parsing characters into non-parsing characters. They will simply become data characters.

  Note: Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

- **PRTSWLST=({LOOP|STOP},SWITCHAFTERINIT,SWITCHAT*time*,**
  {*ddname*1|DYNAMIC=*outclass*},...,{*ddname*n|DYNAMIC=*outclass*})
  was added to the BUILD definition statement to provide 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. The DDNAMEs you specify or request dynamically are used sequentially. The default SYSPRINT file is always used initially (ddname=SYSPRINT).

  Note: Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

- **PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)** operand added to the BUILD definition statement for heartbeat message support. The syntax and suboperand meanings is as follows:

```
|<----------------------------- start time
|
|        |<-------------------- stop time
|        |
|        |        |<----------- frequency
|        |        |
```

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

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 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.

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

- **PVC=(...,*sluname*{-{A|I}},...)** is now allowed to specify an initial state for the named SLU.

  -A (or omitted) is for active and -I is for inactive or idle with active being the default.

  Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

- New **TYPE=DMY REMOTE** was added to improve automation of console command capabilities as well as improvements for the HNAS **PING (XOT) *dmyname*** console command operation.

- **RTEIN=** processing allows **SKIP/*dce-addr*** or **CLEAR/*dce-addr*** to be specified as entries in the RTEIN= list. If SKIP is selected by a DTE address the RTEIN= entry is skipped. If CLEAR is selected by a DTE address then the inbound call is cleared with DIAG=202. The **NAS7701W** alert message has been modified to contain the string 'RTEIN CLEAR' when a RTEIN CLEAR causes the clear.

- The **RTEIN=** and **RTEOUT=** LOCAL parameters now include support for DTE address calling or called address 'wildcards' filtering.

  @ as the <u>first</u> *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. Use of this 'match at end' feature changes the operation of the STRIPRTEIN option -- please see Chapter 4 for details.

  ***** within the *dteaddr* 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.

- **RTEOUT=** processing has been modified to allow a descriptor text value (*desctxt*) to be used in place of the *dteaddr* value in the NAS7717W alarm message. Like *dteaddr*, *desctxt* is limited to 15 characters. This will preserve the format of the NAS7717W message. The new syntax is RTEOUT=(*rmtname*/*dteaddr*{T|S}=*desctxt*,...). *desctxt* can be enclosed in quotes. For example, RTEOUT=(XOTCLNT1/1234T='RMT USER #1',...). When *desctxt* is provided, the NAS7717W message will display <u>DTE IDNT *desctxt*</u> instead of <u>DTE ADDR *dteaddr*</u>.

- **SCHEDULE={*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)}** operand support added to allow scheduled console commands.

As an aid to problem diagnosis, sometimes it is necessary to start/stop traces as well as issue display commands at specific times. Prior to the new SCHEDULE operand, this action could only be done manually via operator intervention. The new SCHEDULE= operand was added to allow commands to be scheduled for automatic execution at defined times.

Syntax: SCHEDULE={*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)}

(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*) is an inline schedule list.

ddname identifies a file that contains a list of times and commands. A *ddname* schedule list can contain the following records:

Comments are allowed and start with an asterisk (*) or semi-colon (;) in record column 1. Comments can also appear on a command line but must start with a semi-colon after the command. A single time and command can be specified on a single record as follows:

hh:*mm*:*ss* , *cmd* ; *comment*

Multiple times and commands can be specified on a single record as follows:

(*hh*:*mm*:*ss* , *cmd* , ... , *hh*:*mm*:*ss* , *cmd*) ; *comment*

Note: The new SCHEDULE= operand support was introduced into 240 as Enhancement APAR 2400064.

- **SOCLMT=*limit*** operand now generates an error message if the sum of the TYPE=XOT REMOTE VCLMT= values (for which the LOCAL is HOME) is greater than *limit*-2. This eliminates a runtime ABEND during HNAS initialization.

  Note: The SOCLMT= operand change was introduced into 240 as Enhancement APAR 2400077.

- **SVC0|4|5=(...,*sluname*[-{A|I}],...)** is now allowed to specify an initial state for the named SLU.

  -A (or omitted) is for active and -I is for inactive or idle with active being the default.

  Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

- **SVC0|5=**{...(sluname/{X*idnum*|*dteaddr*|**<*rmtname*>**}...} permits an MXT|SVC per *dteaddr* and a *dteaddr* via *rmtname*. New **REMOTE TYPE=SVC** was added to provide a smaller MXT footprint without PADPARM=*value* support.

  Note: Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

- **SVC0|5=(...,*sluname*+*gluname*[-{A|I}],...)** is now allowed to specify a generic SLU name (***gluname***) to be associated with a primary HNAS SLU name (***sluname***) for callout resources acquired by a PLU. The ***gluname***, in conjunction with VTAM tables, allows a PLU to acquire an HNAS resource using a generic name.

  Note: *gluname* support was introduced in 240 as Enhancement APAR 2400084.

- **TRCLMT=*count*** expanded from 32767 to 1048575 entries permitting extended tracing capabilities.

Note: Expanded support was introduced into 240 as Enhancement APAR 2400052.

- **TRCTRAP ALRMLIST=(NAS*nnnni*/'*data'*/*data-offset*,...)** operand now accepts ***data*** as quoted string (see TRCTRAP console command enhancement below for more information).

See **Console and Trace Enhancements** for **PARM=** Start Parameter Changes & Improvements

Alert/Alarm Message and Clear/Reset Code Enhancements:

- **NASC*nnni*** console command response messages were added to improve console command decode processing. Please refer to the 'new for 240' references on the specific entries in the Console Messages section of the Messages and Codes guide for additional information.

- **NASC052W** console command response message issued if EXEC LIST is specified without command list (*ddname* or (*cmd*1,...,*cmd*n).

NASC052W EXEC COMMAND LIST WAS NOT PROVIDED, REQUIRED FOR EXEC COMMAND

Note: NASC052W error message was introduced into 240 as Enhancement APAR 2400034.

Note: NASC052W error message was retired from 240 as part of Enhancement APAR 2400098.

- **NASC053W** console command response message issued if EXEC *ddname* identifies an empty command list file.

NASC053W EXEC *ddname* FILE IS NULL, EXEC COMMAND ABORTED

Note that a command list file is considered empty if it contains no commands or display comments (# text) but may contain non-display comments ('* text').

Note: NASC053W error message was introduced into 240 as Enhancement APAR 2400034.

- **NASC100E** console command response message has been modified to more completely identify why a command that operates on the LNM=, RNM=, LUNM= or ID= modifiers is aborted when none of these modifiers are specified.

```
NASC100E ID= AND OTHER MODIFIERS OMITTED, REQUIRED FOR cmdname COMMAND 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)
```

Note: The new NASC100E message change was introduced into 240 by Enhancement APAR 2400098.

- **NAS003*ni*** alert messages now identify the type of shutdown as well as the actual end status.

NAS0030I SHUTDOWN STARTED, RELEASING RESOURCES

is issued when QS, QQ/*password* or QE *ddname* are entered (no SNAP dump is taken). The is a fast shutdown.

NAS0031I SHUTDOWN STARTED, FORMATTING STORAGE, RELEASING RESOURCES

is issued when QY/*password* is entered (SNAP dump is taken). The SNAP dump extends the shutdown processing and adds to SYSPRINT content.

`NAS0032I SHUTDOWN STARTED, FORCED ABEND REQUESTED`

is issued when QA/*password* is entered (ABEND dump is taken). The ABEND dump extends the shutdown processing and adds to SYSPRINT content. It should only be used when requested by Comm-Pro.

`NAS0035I SHUTDOWN COMPLETED, RC=xx`

is issued when shutdown completes.

- **NAS01*nni*** alarm messages are now generated when the SHOW ON|OFF|ERR ... MORE|LESS ... console command is executed.

  For ON

  `NAS0110I SYSCONS ERROR AND INFO ALARM OUTPUT ENABLED BY` ***`consname`***

  For OFF

  `NAS0111W SYSCONS NON-FORCED ALARM OUTPUT DISABLED BY` ***`consname`***

  For ERR

  `NAS0112I SYSCONS ERROR ALARM OUTPUT ENABLED BY` ***`consname`***

  For MORE|LESS

  `NAS0130W SYSCONS ALARM OPTIONS MODIFIED BY` ***`consname`***

  Note: These messages are only issued if a change is effected by the command, for example from the ON to OFF state. The messages are withheld if the command does not effect a change, for example when ON is specified and the ON state is already active (effectively, a no operation).

  Note: The new NAS01*nni* alarm messages were added to 240 as part of Enhancement APAR 2400017.

- **NAS020*ni*** alert messages are now generated when the new PRTSWLST= SYSPRINT switching support is enabled providing SYSPRINT status notification:

  `NAS0205I` ***`sysprint`*** **`LOGGING TERMINATED`**

  This message is issued after the SYSPRINT log file identified by DDNAME *sysprint* has been closed during SHUTDOWN processing. This message is issued unconditionally even when PRTSWLST= support is not being used.

  `NAS0207W ALL PRTSWLST DDNAMES HAVE BEEN USED, LOGGING TERMINATED`

  This message is issued after the last SYSPRINT log file in the PRTSWLST= operand list has been closed and the stop action is in effect (PRTSWLST=(STOP,...) was specified). The PRNT OPEN *ddname* console command will have to be used to manually open a new SYSPRINT log file.

Note: This message will not be issued if the LOOP action is in effect (PRTSWLST=(LOOP,...) was specified).

NAS0208I *sysprint* LOG FILE HAS BEEN ACTIVATED FOR LOGGING

This message is issued when the SYSPRINT log file identified by DDNAME *sysprint* has been opened automatically as the next log file in the PRTSWLST= operand list.

NAS0209E *sysprint* LOG FILE COULD NOT BE OPENED

This message is issued when the SYSPRINT log file identified by DDNAME *sysprint* could not be opened. If alternate SYSPRINT file(s) have been defined via the PRTSWLST= operand on the BUILD definition statement, a new SYSPRINT log file will be opened automatically. Otherwise, the PRNT OPEN *ddname* console command will have to be used to manually open a new SYSPRINT log file.

- **NAS02*nni*** alarm messages are now generated when the PRNT ON|OFF ... DATE ON|OFF ... console command is executed.

  For ON

  NAS0210I SYSPRINT LOGGING ENABLED BY *consname*

  For OFF

  NAS0211W SYSPRINT LOGGING DISABLED BY *consname*

  For DATE ON|OFF, QLLC ON|FF, TCP ON|OFF ...

  NAS0230W SYSPRINT LOGGING OPTIONS MODIFIED BY *consname*

  Note: These messages are only issued if a change is effected by the command, for example from the ON to OFF state. The messages are withheld if the command does not effect a change, for example when ON is specified and the ON state is already active (effectively, a no operation).

  Note: The new NAS02*nni* alarm messages were added to 240 as part of Enhancement APAR 2400017.

- **NAS0210I** alert messages are now generated when a PRTSWLST= event occurs:

```
NAS0210I PRTSWLST SWITCHNOW REQUESTED BY  consname
NAS0210I PRTSWLST SWITCHAFTERINIT ACTION INITIATED
NAS0210I PRTSWLST SWITCHAThh00 ACTION INITIATED
NAS0210I PRTSWLST SWITCH ON FULL DATASET INITIATED
NAS0210I PRTSWLST SWITCH ON PRTLMT INITIATED
```

  These messages are issued when the a switch occurs in the PRTSWLST= operand of the BUILD definition statement. The text indicates why the SYSPRINT log file switch was requested. consname identifies the name of the console that issued the PRNT command. For the local console (SYSCONS), this is WACONPCE. For a remote console, this is RCONnnnn where nnnn is the remote console number.

- **NAS022*ni*** alert messages are now generated when the new **PRNT {OPEN|CLSOPN}** console command is used to open or close SYSPRINT files:

  NAS0220W *sysprint* LOG FILE OPENED BY *consname*

  This message is issued when the PRNT {OPEN|CLSOPN} *sysprint* console command is used to open a new SYSPRINT file.

  sysprint is the new SYSPRINT DDNAME that was opened. *consname* identifies the name of the console that issued the PRNT command. For the local console (SYSCONS), this is WACONPCE. For a remote console, this is RCON*nnnn* where *nnnn* is the remote console number.

  NAS0221W *sysprint* LOG FILE CLOSED BY consname

  This message is issued when the **PRNT {CLOSE|CLSOPN** newddname} console command is used to close an old SYSPRINT file.

  sysprint is the old SYSPRINT DDNAME that was closed. *consname* identifies the name of the console that issued the PRNT command. For the local console (SYSCONS), this is WACONPCE. For a remote console, this is RCON*nnnn* where *nnnn* is the remote console number.

- **NAS0299I** messages generated for **PULSE** heartbeat message. See description of new PULSE= operand above for additional information.

- **NAS0310W** alert message is issued when the VARY console command is used to change a resource state.

  NAS0310W VARY ACTION=*action* TYPE=RMT RNM=*rmtname* BY *consname*
  NAS0310W VARY ACTION=*action* TYPE=LCL LNM=*lclname* BY *consname*
  NAS0310W VARY ACTION=*action* TYPE=LU LUNM=*sluname* BY *consname*

  The ***action*** value can be ON or OFF. ***rmtname***, ***lclname*** or ***sluname*** identify the name of RMT, LCL or LU resource be varied ON or FF. ***consname*** identifies the name of the console that issued the VARY command. For the local console (SYSCONS), this is WACONPCE. For a remote console, this is RCON*nnnn* where *nnnn* is the remote console number.

- **NAS1*n*11E** messages are now issued when an incorrectly formatted table is specified for the USSTAB=, LOGTAB=, or OPTIONS=NRITAB= operand. For example:

  NAS1311E REMOTE *mchname* LOGTAB=ISTINCDT IS NOT A VALID LOGON TABLE, REQUIRED

  Note: New NAS1*n*11E message support was introduced into 240 as Enhancement APAR 2400032.

- **NAS1154E** message is issued when SCHEDULE=*ddname* queue limit is exceeded.

  NAS1154E BUILD SCHEDULE *ddname* COMMAND QUEUE LIMIT *count/limit*
  EXCEEDED, REQUIRED

  The ***ddname*** value identifies the DDNAME of the file that is too large. ***count*** represents the number of bytes required for the file and ***limit*** represents the maximum number of bytes allowed for any file.

  Note: The new SCHEDULE= operand support was introduced into 240 as Enhancement APAR

2400064.

- **NAS1301D** message is issued when an SLU initial value is omitted for an SLU named in the LUNAME=, PVC=, SVC0=, SVC4= or SVC5= operand. For example:

```
NAS1301D REMOTE operand SLU INIT VALUE (-{A|I}) OMITTED FOR ONE OR MORE
NAS1301D       ENTRIES STARTING WITH operand XXXX, -A (ACTIVE) ASSUMED
```

Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

- **NAS1311S** message is now issued when the sum of the TYPE=XOT REMOTE VCLMT values (for which the LOCAL is HOME) is greater than SOCLMT=*limit* value minus 2. The new NEEDED=*count* value was added to show how many additional sockets are required.

```
NAS1311S REMOTE XOTCNOT2 HOME XOTSRVR LIMIT REACHED
                SOCCNT=00303 SOCLMT=00602 NEEDED=00301
```

This message is issued when NEEDED+SOCCNT>SOCLMT. SOCCNT includes 2 reserved sockets.

Note: The NAS1311S message change was introduced into 240 as Enhancement APAR 2400077.

- **NAS1*nnni*** configuration messages updated in an effort to provide improved configuration error messages troubleshooting as well as adding messages for new 240 CDF parameters and options

- **NAS2109S** alarm message is now issued when an ***unexpected TCPIP interrupt*** is presented and the DBUG TCP start parameter is in effect.

```
NAS2109S CLIENT=010.117.056.100(04545) SOCKID=0001 PCEID=000C NAME=R1CNIN
NAS2109S UNEXPECTED TCPIP INTERRUPT, IGNORED
NAS2109S IPARM=00018502000000024001300FAD5C1E2E3F0E3E2D6
               0000D740004EA808004EAB900000000000508C00
               000000001000000001348352413483533
```

When DBUG TCP is not active, the following 0198 ABEND will occur when an unexpected TCPIP interrupt is presented:

```
HALT AT LOC 80065CB4 IN NASUTIL : TCPIP INTERRUPT UNEXPECTED
```

- **NAS2110S** alarm message is now issued when an ***invalid TCPIP reply ID*** is detected and the DBUG TCP start parameter is in effect.

```
NAS2110S SERVER=172.029.127.220(01998) SOCKID=0000 PCEID=0009 NAME=LXOT
NAS2110S INVALID TCPIP INTERRUPT REPLY ID, IGNORED FOR SELECT
NAS2110S EXPECTED=0000001D PRESENTED=0000001C
NAS2110S IPARM=000185020000001C001300FAD5C1E2E3F0E3E2D6
               0000D740004E8C18004E8FA000000000005187E0
               00000000000000001209175512101732
```

When DBUG TCP is not active, the following 0198 ABEND will occur when an invalid TCPIP reply ID is presented:

```
HALT AT LOC 80070952 IN NASTCP  : TCPIP REPLY ID FAILURE
```

- **NAS2152E alarm** message is now issued when a CANCEL command timeout occurs.

```
NAS2152E CLIENT=010.117.056.100(04545) SOCKID=0001 PCEID=000C NAME=R1CNIN
NAS2152E CANCEL REQUEST INTERRUPT LOST, NORMAL  COMPLETION ASSUMED FOR CONNECT
```

- **NAS2502E** message now displays **CLOSEONTAPFAILURE** when a router contact is lost and the CLOSEONTAPFAILURE option is in effect. This is the default option when NOCLOSEONTAPFAIL-URE is not specified in the CDF.

- **NAS2505E** message was added and displays **NOCLOSEONTAPFAILURE** when a router contact is lost and the NOCLOSEONTAPFAILURE option is in effect.

  Note: [NO]CLOSEONTAPFAILURE support was introduced into 240 as Enhancement APAR 2400055.

- **NAS2511M** message that is issued for **MON TAP** now contains a retry count (*ddddd*). The count is reset when a successful TAP response is received. The count indicates the number of unsuccessful TAP attempts.

```
NAS2511M CLIENT=010.117.056.100(02704) SOCKID=0001 PCEID=000B NAME=R1CNIN
NAS2511M XOT TAP TIMEOUT, RESPONSE NOT RECEIVED FOR CONNECTION SETUP (ddddd)
```

  Note: Support was introduced into 240 as Enhancement APAR 2400055.

- **NAS251*n*M** messages that are issued for **MON TAP** console command activity are normally logged in SYSPRINT only can now (optionally) be displayed on the SYSCONS using the ALRMFLTR= operand (FC) or (FU) suffixes.

```
NAS2513M CLIENT=010.117.056.100(02190)
         SOCKID=0002 PCEID=000B NAME=R1CNIN
NAS2513M XOT TAP SEQUENCE IS STARTING,
         TRANSMITTING CALL REQUEST
NAS2513M PKT=0000001010010B000001000000C8D5C1E2E3C1D7

NAS2515M CLIENT=010.117.056.100(02190)
         SOCKID=0002 PCEID=000B NAME=R1CNIN
NAS2515M XOT TAP SEQUENCE IN PROGRESS,
         RECEIVED CLEAR REQUEST
NAS2515M PKT=000000051001130342

NAS2513M CLIENT=010.117.056.100(02190)
         SOCKID=0002 PCEID=000B NAME=R1CNIN
NAS2513M XOT TAP SEQUENCE IN PROGRESS,
         TRANSMITTING CLEAR CONFIRM
NAS2513M PKT=00000003100117

NAS2517M CLIENT=010.117.056.100(02190)
         SOCKID=0002 PCEID=000B NAME=R1CNIN
NAS2517M XOT TAP SEQUENCE COMPLETED SUCCESSFULLY
```

- **NAS251*n*M** messages that are issued for **MON TAP** can now provide information similar to **PING** console command activity as requested by user via the **PKTDATA|MAXDATA|MINDTA|NODATA** arguments. For **MAXDATA**:

```
NAS2513M CLIENT=010.117.056.100(02190)
         SOCKID=0002 PCEID=000B NAME=R1CNIN
NAS2513M XOT TAP SEQUENCE IS STARTING,
         TRANSMITTING CALL REQUEST
NAS2513M DTEADDR=ddd...ddd DCEADDR=ddd...ddd
NAS2513M FAC=xx...xxx
NAS2513M CUD=xx...xxx
```

Note: NAS251*n*M DTEADDR=, etc. display support was introduced into 240 as Enhancement APAR 2400044.

- **NAS261*n*M** messages are now issued for **PING** console command activity will be logged in SYSPRINT and (optionally) displayed on the SYSCONS using the ALRMFLTR= operand (FC) or (FU) suffixes.

```
NAS2613M PINGING SOCKET=aaa.bbb.ccc.ddd(port) FOR consname
NAS2613M         DTEADDR=ddd...ddd DCEADDR=ddd...ddd
NAS2613M         FAC=xx...xxx
NAS2613M         CUD=xx...xxx

NAS2611M PING TIMED OUT FOR consname

NAS2615M PING COMPLETED FOR consname, CLEAR REQUEST
         RECEIVED, CAUSE/DIAG=ccdd

NAS2617M PING COMPLETED FOR consname, CALL ACCEPT
         RECEIVED
```

- **NAS261*n*M** messages have been updated for **PING** console command to display a sequence number so that termination messages (NAS2611M, NAS2615M, NAS2617M) can be easily correlated to initiation messages (NAS2613M). In addition, termination messages now display the target IP and DTE addresses. The new message formats are as follow:

```
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

NAS2611M PING CALL TIMED OUT FOR SOCKET=aaa.bbb.ccc.ddd(port)
         DTEADDR=ddd...ddd (seqno)

NAS2615M PING CALL CLEARED (cc/dd) FOR SOCKET=aaa.bbb.ccc.ddd(port)
         DTEADDR=ddd...ddd (seqno)

NAS2617M PING CALL ACCEPTED FOR SOCKET=aaa.bbb.ccc.ddd(port)
         DTEADDR=ddd...ddd (seqno)
```

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.

- **NAS1391I** messages now issued to display MXT operand overrides. When a user associates an MXT with a PVC= or SVC0/3/5= operand entry, he is asking HNAS to override operands on the root MCH with those from the MXT. In some cases, this can lead to confusion if one does not remember to review the HNAS documentation to see which operands can be overridden based on the associated resource type.

  The configuration logic has been modified to display, for each SLU or SPU defined in the PVC= or SVC0/3/5= operands with an associated MXT (or SVC), those MXT operands that will override operands of the same name on the root MCH. Not all MXT operands are valid overrides for all SLUs or SPUs. For example, IDLETO= is a valid override when specified for an MXT that is associated with a SVC0/5= SLU but is not used for a PVC SLU.

  MXT override messages are listed at the end of the CDF scan in NAS1391I messages. Only those MXT operands that are specified and, of these, only those that are valid as overrides for an SLU or SPU are displayed. If an MXT operand is not listed, it is because it was either omitted from the MXT (an hence cannot override the MCH operand) or, if specified, is not valid as an override for the SLU or SPU resource. See expanded content for additional information.

  Note: NAS1391I message support was introduced into 240 as Enhancement APAR 2400018.

- **NAS3796I** alert message has been added to indicate that a Fast Connect GATE CTCP has sent HNAS a call accept packet to the named HNAS SLU. This indicates the start of a GATE FC session initiated by an inbound call.

  NAS3796I FC LU *luname* ON MCH *mchname* REC'D CALL ACCEPT FROM *pluname* FOR SESSION WITH *ip-addr(port)*

- **NAS3797I** alert message is now generated when a GATE control session is bound. Previously this alert was only used for BINDs received for PVC LUs. This will eliminate the requirement to view the HNAS DLU command display output to confirm that the GATE control session is active.

  NAS3797I LU *lu-nm* RECEIVED BIND FROM PLU *plu-nm*

  Note: This logic change was introduced into 240 under APAR 2400005.

- **NAS5705W** RESET SCHEDULED alert message has been modified to provide **DIAGX=0000|0001** values in an effort to better define the cause of the PVC reset. The HNAS Reset Diagnostic code table entry 196 (C4) now contains definitions for these new values.

  Note: This logic change was introduced into 240 under APAR 2400053.

- **NAS5723W** messages issued for IDTST= ID String length exception, sample follows:

  NAS5723W DATAFONO IDTST LU *lu-name* REC'D ID STRING W LENGTH > 11

  Note: This logic change was introduced into 240 as part of Enhancement APAR 2400074.

- **NAS5724W** messages issued for IDTST= 'M' messages response not received in 'nn' seconds, sample follows:

  NAS5724W LU *lu-name* DID NOT RECEIVE 'M' MSG SYNC RESP FROM REMOTE

Note: This logic change was introduced into 240 as part of Enhancement APAR 2400074.

- **NAS5725W** messages issued for IDTST= Bad ID condition, sample follows:

  NAS5725W LU lu-name DISCARDING MSG W BAD ID FROM PLU plu-name

  Note: This logic change was introduced into 240 as part of Enhancement APAR 2400074.

- **NAS5726W** messages issued when EMSGE option was specified for a Datafono session and the 16 second PLU response timer expired, sample follows:

  NAS5726W EMSGE MESSAGE SENT TO LU *lu-name* (16 SEC PLU DELAY)

  Note: This logic change was introduced into 240 as part of APAR 2400097.

- **NAS7701W** alert message has been modified to contain the string 'RTEIN CLEAR' when a RTEIN CLEAR causes the clear. This change was implemented to support the new **RTEIN=** processing which allows **SKIP/*dce-addr*** or **CLEAR/*dce-addr*** to be specified as entries in the RTEIN= list.

  NAS7701W CALL FROM *iii.iii.iii.iii*(*port*) CAN'T ROUTE CALLED ADDR
  ***lddddddd dddddddd* [RTEIN CLEAR]**

- **NAS7716W** this GATE call request failure alert message will display as a single line alert if OPTIONS=ALRMSGTXT=SHORT is coded on BUILD (APAR 2400068).

- **NAS7717W** alert message will now display the descriptor text value (*desctxt*) instead of the *dteaddr* value when *desctxt is* coded on the RTEOUT=(*rmtname*/*dteaddr*/*desctxt*,...) operand.

  NAS7717W LU *sluname* CALL TO DTE IDNT *desctxt* VIA REMOTE *rmtname* FAILED

  This call request failure alert will display as a single line alert message if OPTIONS=ALRMS-GTXT=SHORT is coded on BUILD (APAR 2400068).

- **NAS7718T** this trace message and 3 **NAS7798T** messages are recorded in SYSPRINT when TRC-MCH ICR ON is set and a call request is received. When OPTIONS=ALRMSGTXT=SHORT is coded on BUILD (APAR 2400068) then the NAS7718T and NAS7798T messages are replaced with a **NAS7730I** message which is sent to the operator's console and SYSPRINT. This allows job log monitoring programs to see inbound call activity.

- **NAS7719T** this trace message and 3 **NAS7798T** messages are recorded in SYSPRINT when TRC-MCH OCR ON is set and a call request is sent by HNAS. When OPTIONS=ALRMSGTXT=SHORT is coded on BUILD (APAR 2400068) then the NAS7718T and NAS7798T messages are replaced with a **NAS7731I** message which is sent to the operator's console and SYSPRINT. This allows job log monitoring programs to see outbound call activity.

- **NAS7730I** this single line message records an inbound call request on the operator's console and SYSPRINT when TRCMCH ICR ON is active and OPTIONS=ALRMSGTXT=SHORT is coded on BUILD.

- **NAS7731I** this single line message records an outbound call request on the operator's console and SYSPRINT when TRCMCH OCR ON is active and OPTIONS=ALRMSGTXT=SHORT is coded on BUILD.

- **NAS7798T** these messages follow NAS7718T or NAS7719T messages when TRCMCH ICR/OCR ON is active an OPTIONS=ALRMSGTXT=LONG (the default) is coded on BUILD. If ALRMSGTXT=SHORT is coded these messages do not appear in SYSPRINT.

- **NAS92*nns*** messages added for additional authorization processing (please see HNAS Messages and Codes documentation for details).

- **NAS9205I** message is now withheld until today's date is within 60 days of the EOTDATE. In addition, the message severity is changed from 'I' to 'S' as today's date gets closer to the EOTDATE. Specifically, the message ID is generated as follows:

  `NAS9206I`, when TODAYSDATE >= EOTDATE-60
  `NAS9206W`, when TODAYSDATE >= EOTDATE-45
  `NAS9206E`, when TODAYSDATE >= EOTDATE-30
  `NAS9206S`, when TODAYSDATE >= EOTDATE-15

  Note: The new NAS9205I message change was introduced into 240 by Enhancement APAR 2400106.

- **NAS9206I** message is now withheld until today's date is within 60 days of the EOMDATE. In addition, the message severity is changed from 'I' to 'S' as today's date gets closer to the EOMDATE. Specifically, the message ID is generated as follows:

  `NAS9206I`, when TODAYSDATE >= EOMDATE-60
  `NAS9206W`, when TODAYSDATE >= EOMDATE-45
  `NAS9206E`, when TODAYSDATE >= EOMDATE-30
  `NAS9206S`, when TODAYSDATE >= EOMDATE-15

  Note: The new NAS9206I message change was introduced into 240 by Enhancement APAR 2400098.

- HNAS **Clear Request** diagnostic codes **D8 (217)** and **D9 (218)** now provide special **DIAGX=** values in alert messages that replace validity checks in V2R4M0. Please contact customer support if one of these CAUSE/DIAGX combinations is seen.

  D8 (217) Clear diagnostic code - New **DIAGX= codes 17-1A**:

  17 VTAM SEND routine entered with RPL already busy.
  18 VTAM MODCB macro failed.
  19 SEND routine passed QLLC message with invalid FID byte.
  1A SEND routine passed invalid QLLC LUSTAT request.

  D9 (218) Clear diagnostic code - New **DIAGX= codes 05-0B**:

  05 VTAM RECEIVE routine entered with RPL already busy.
  06 Normal flow request value from exit routine invalid.
  07 VTAM RECIVE routine entered, required buffer not present.
  08 VTAM RECEIVE completion entered, RPL flags do not indicate reason.
  09 VTAM RECEIVE completion routine entered, LUHNFREQ indicates active command.
  0A VTAM RECEIVE completion routine entered, RPL does not show completed command.
  0B VTAM RECEIVE RESPONSE completion routine entered RPL does not show cmd for response

- HNAS **Clear Request** diagnostic codes **DF (223)** now provide new **DIAGX=** codes **08|09** values:

08 ‘M’ message sync response not received from remote.
09 PLU sent data to pseudo leased LU with no VC session.

Note: This logic change was introduced into 240 as part of APAR 2400097.

Console, Trace and Debug Enhancements (Commands, Start Parameters...):

- When # is the first character of a console input record, the record is not processed as a command but is simply echoed after the HNASCMD-> prefix (for example, HNASCMD-> # *text*). The comment record may be specified like any other console input, that is, via remote command line input, the modify interface (/F *jobname*,# *text*), the CONCMDQ=(...,# *text*,...) operand or embedded in an EXEC *cmdlist* file.

  Note: Console # comment processing was added to 240 as part of Enhancement APAR 2400017.

- **All** commands now echo their invocation parameters following the HNASCMD-> display prefix and their execution parameters following the HNASXEQ-> display prefix. The execution parameters include specified and default modifiers and parameters. For example, if **TRCMCH MCH1 ON ICR** is entered, the following lines are echoed back to the console operator:

  ```
  HNASCMD-> TRCMCH MCH1 ON ICR                 <- echo entered command
  HNASXEQ-> TRCMCH RNM=MCH1 ON GBL ICR ON      <- show execution parameters
  ```

  For additional information on these trace commands, please see Console Subsystem documentation or command HELP displays.

  Note: HNASXEQ-> support was introduced into 240 as Enhancement APAR 2400014.

- **All** commands that force another command to be executed now enqueue the second command in a LIFO rather than FIFO fashion so that the propagated command is executed immediately after the original command (for example, DMCH FMF3 -> DLU, MLCL *lclname* INIT={ACTIVE|IDLE) -> VARY *lclname* {ON|OFF}, MLCL *rmtname* INIT={ACTIVE|IDLE) -> VARY *rmtname* {ON|OFF})

  Note: LIFO console command enqueue support was introduced into 240 as Enhancement APAR 2400028.

- **LCLCONS** or ***rconname*** 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. These new arguments are provided for commands that operate on console PCEs.

  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. Only consoles that are in the active state can be targeted by the

rconname parameter.

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

- **LCLCONS**, **NASUTIL**, **PING**, **TIMER**, **XOTUTIL**, **XTPUTIL** or ***pcename*** 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. These new arguments are provided for commands that operate on PCEs.

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: LCLCONS, NASUTIL, PING, TIMER, XOTUTIL, XTPUTIL and *pcename* argument support was introduced into 240 with Enhancement APAR 2400075.

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

`EXECCMD->` `=>` command came from CONCMDQ= operand or EXEC command.
`SCHDCMD->` `=>` command came from 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.

- **ALARM FILTER=** now supports (FC) and (FU) message ID suffixes to allow forced conditional (subject to SHOW state) and forced unconditional (SHOW state ignored) SYSCONS display. For example, if ALARM FILTER=(...,NAS2020I(FC),NAS0299I(FU),...) is specified, NAS2020I alarm messages will be sent to SYSCONS except when SHOWOFF is effect while NAS0299I alarm messages will be sent to SYSCONS regardless of the SHOW state.

- The **ALARM LOG=?** command now displays date and time of last alert message log activity for each alarm message entry. This information is very useful when immediate access to SYSPRINT is not available.

```
ALARM LOG=?

ALARM_ID  TOTAL_CT    LOG_DATE  LOG_TIME   (0013/0256)
NAS0001I  00000001  2006/02/21  17:08:03
NAS0070W  00000001  2006/02/21  17:08:00
:
```

- The **ALARM LOG=?** command is now executed 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*.

- The **ALARM MSGTXT={?|LONG|SHORT}** command support has been added to allow the BUILD OPTIONS=ALRMSGTXT= operand to be displayed and/or toggled. This OPTION allows you to select the long (default) or short 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 (see NAS773*x*I messages above for additional information).

  When ALARM MSGTXT=? is entered, the current ALRMSGTXT= operand value is displayed.

  When ALARM MSGTXT=LONG is entered, the alarm message format is set to long (old format).

  When ALARM MSGTXT=SHORT is entered, the alarm message format is set to short (new format).

  Note: ALARM MSGTXT= command support was introduced into 240 as Enhancement APAR 2400068.

- The **ALARM PULSE={?|*|(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*)}** command support has been added to allow dynamic heartbeat message control.

  When ALARM PULSE=? is entered, the current PULSE= operand values are displayed.

When ALARM PULSE=* is entered, the current PULSE= operand values are reset. Pulsing is stopped.

When ALARM PULSE=(*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,*seconds*) is entered the syntax meaning is as follows:

```
                 |<---------------------------- start time
                 |
                 |        |<-------------------- stop time
                 |        |
                 |        |        |<----------- frequency
                 |        |        |
                 V        V        V
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 savearea. 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.

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

- The **ALARM LCLCONS|*rconname action*** command support added to allow one console to modify alarm parameters for another console.

```
action = {ON|OFF} {TEST='text'}
         {FILTER={?|*|({A|P|S},id1({A|P|S|FC|FU}),...,idn({A|P|S|FC|FU}))}}
         {LIMITS={?|*|@|(tsec,ict,dct,act,wct,ect,sct,uct)}}
         {LOG={?|CLEAR|RESET}}
         {MSGTXT={?|SHORT|LONG}}
         {PULSE={?|*|(hh:mm:ss,hh:mm:ss,seconds)}}
         {SHOW}
```

Note: The ALARM LCLCONS|*rconname action* change was introduced into 240 with Enhancement APAR 2400075.

- **BFR** (internal) trace record now provides buffer allocation and release timestamp and tracestamp which will aid in diagnosing some types of problems. It is advantageous to know when a buffer was allocated and when it was released. Using the allocation timestamp and release timestamp, it can be determined how long a buffer has been in use. The current trace table pointer is also saved at allocation and release so these events can be correlated with trace activity.

Note: This change was introduced into 240 as Enhancement APAR 2400051.

- **DLU** console command now displays the new LUTO column to show the active LU timer in effect (blank implies that no timer is running).

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

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

- **DVC** console command now displays the new VCTO column to show the active VC timer in effect (blank implies that no timer is running).

  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.
  RCWT - PVC reconnect wait state (PVCRECONTMR=), REQSESS issued on timer expiration.
  REXP - QLLC PIU sent to SPU, response expected.
  RQSD - REQSHUT sent to VTAM, SHUTDOWN expected.
  RQSE - REQSESS sent to VTAM, BIND expected.
  STLG - REQSESS wait state (REQSESSDELAY=), 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 (PVCSETUPTMR=), Setup issued on timer expiration.
  TEST - QLLC QTEST sent to SPU, response expected.
  XID - QLLC QXID sent to SPU, response expected.

  Note: The DVC VCTO change was introduced into 240 with Enhancement APAR 2400078.

- The **DLU** and **DVC** console commands will now display the VC setup direction in the new **SESSINIT** column (old CID column) as **INBOUND** or **OUTBOUND**. This works for SVCs and PVCs. Blanks are displayed in the SESSINIT column if no VC connection is active. The CID can be displayed if the SHOWCID operand is provided for DLU and DVC.

- The **DLU** and **DVC** console commands will now display a ‘P’ in place of the leading zero for a PVC VCN value. The VCN for a PVC can range from 0 to 255 while the VCN display ranges from 0 to 4095 to accommodate both PVCs and SVCs.

  Note: The PVC change for the DLU and DVC console commands was introduced into 240 as Enhancement APAR 2400028.

- The **DLU**, **DPCE** and **DVC** console commands will now display header information that is appropriate for both XOT and XTP resources. In the past, only XTP header information was displayed. For example, the DVC header will now look as follows:

```
IFN|
```

```
MPID RPID VCN  SESSINIT SLUNAME  VCOPT VCST VCTO  LLC CLGADDR
```

The MPID column identifies the PCE ID for the MCH that owns a VC. The IFN|RPID column identifies either the Interface Number for an XTP MCH or the PCE ID for the XOT REMOTE over which the VC connection is active.

Note: This new display header support was introduced into 240 as Enhancement APAR 2400014.

- The **DNAS** command has been modified to display additional customer specific content. These include the **DNAS COMMAND ENTERED AT hh:mm:ss ON 2006/07/31** entry depicting the time the DNAS command was entered, **CUSTID='*cid_cust#*'** display line, **CUSTINFO='*text*'** display line containing up to 32 characters of customer account information while the **MAINTENANCE/USE ANNIVERSARY DATE IS *yyyy/mm/dd*** display line contains the HNAS license maintenance/use coverage period. **DATAFONO SUPPORT IS INCLUDED** will display only when the distribution is enabled for Datafono support.

```
HNAS --> VERSION=V2R4M0 DIST=NON-SMP
         ASMDATE=2006/07/31 ASMHOST=ZOS
         RUNNING UNDER z/OS 01.04.00
         DNAS COMMAND ENTERED AT 11:22:12 ON 2006/07/31 <-- new line
         HNAS PROGRAM STARTED AT 11:22:08 ON 2006/07/31
         HNAS PRODUCT CREATED AT 06:36:58 ON 2006/07/31
         CREATED WITH MAINTENANCE THROUGH APAR 2400000
         MOST RECENT MAINTENANCE APPLIED IS APAR 2400000
         SHIPID=0000000000199999 AUTH=00
         CUSTMAC=                                       <-- if present
         CUSTOBJ=                                       <-- if present
         CUSTID=cid_cust#                               <-- if present
         CUSTINFO=optional cust information             <-- if present
         MAINTENANCE/USE ANNIVERSARY DATE IS 2007/07/31 <-- new line
         DATAFONO SUPPORT IS INCLUDED                   <-- if present

         APARID   MODULE  (APPLIED MAINTENANCE)
         ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400000
```

- The **DNAS** console command now supports **DNAS JCL|DDNAMES** parameter to display the running HNAS JCL DDnames. This display is very useful when DDname information is required by remote console users who do not have access to SYSPRINT. Please refer to the sample DNAS JCL output and description located in the Console SubSystem documentation.

- The **DNAS** console command has been modified to display the ‘installed under’ system and date stamp in addition to ‘created under’ and ‘running under’ system and date stamp. The new display appears as follows:

```
HOST NAS INFORMATION FOLLOWS
     ASMDATE=2010/09/16 ASMHOST=ZOS                <--- deleted line
     HNAS VERSION=V2R4M0 DIST=NON-SMP
     HNAS PROGRAM RUNNING UNDER z/OS 01.10.00
     HNAS PRODUCT INSTALLED UNDER z/OS 01.10.00         <-- new line
     HNAS PRODUCT CREATED UNDER z/OS 01.11.00
     DNAS COMMAND ENTERED AT 07:33:27 ON 2010/09/16
     HNAS PROGRAM STARTED AT 07:33:26 ON 2010/09/16
     HNAS PRODUCT INSTALLED AT 07:29:00 ON 2010/09/16   <-- new line
     HNAS PRODUCT CREATED AT 07:08:18 ON 2010/09/16
```

```
HNAS PRODUCT CREATED WITH MAINTENANCE THROUGH APAR 240BETA
MOST RECENT MAINTENANCE APPLIED IS APAR 2400103
AUTH=000  SHIPID=1100000011199999
CUSTID=SFD_99999                                 <-- if present
CUSTINFO=COMM-PRO ASSOCIATES                     <-- if present
MAINTENANCE/USE ANNIVERSARY DATE IS 2010/09/31
DATAFONO SUPPORT IS INCLUDED                     <-- if present
CUSTMAC=                                         <-- if present
CUSTOBJ=                                         <-- if present


ALL MAINTENANCE ON THROUGH MOST RECENT APAR 2400103
```

Note: Support was introduced into 240 as Enhancement APAR 2400103.

- The **DNWDF** console command has been added to display the NEWDEFN CDF that is maintained in memory while HNAS is running. The NEWDEFN CDF can be modified by the MLCL and MRMT commands. The NEWDEFN CDF is written to disk when HNAS is SHUTDOWN. The GENNWDF start parameter is required for NEWDEFN CDF support. Please refer to the DNWDF command description located in the Console SubSystem documentation for more information.

- The **DRMT** command now displays **[NO]CLOSEONTAPFAILURE** option for XOT|XTP REMOTEs.

  Note: Support was introduced into 240 as Enhancement APAR 2400055.

- The **DRMT** command now displays **[NO]CUD0SELECTSLU** option for MCH|MXT REMOTEs.

  Note: Support was introduced into 240 as Enhancement APAR 2400074.

- The **DRMT** command now displays **[NO]DELAYBINDRESP** option for MCH REMOTEs.

- The **DRMT** command now displays **[NO]IDTST** option for DFX REMOTEs.

  Note: Support was introduced into 240 as Enhancement APAR 2400074.

- The **DRMT** command now displays **PVCRECONTMR=**, **PVCSETUPTMR=** and **PVCSETUPREJ=** options for MCH|MXT REMOTEs.

  Note: Support was introduced into 240 as Enhancement APAR 2400059.

- The **DRMT** command now displays **[NO]RESETINO** option for MCH REMOTEs.

  Note: Support was introduced into 240 as Enhancement APAR 2400081.

- The **DRMT** command now displays **SVCCALLTMR=** option for MCH|MXT REMOTEs.

  Note: Support was introduced into 240 as Enhancement APAR 2400069.

- The **DRMT** command now displays the generic SLU name if one was specified for the **SVC0|5=** operand for MCH REMOTEs (e.g., SVC0=(...,*sluname+gluname*/...).

  Note: *gluname* support was introduced into 240 as Enhancement APAR 2400084.

- The **DRMT** command has been modified to display the time and date when a table specified for the LOGTAB=, USSTAB= and OPTIONS=NRITAB= operands is loaded into memory. This means that you will be able to tell if a table was loaded during the CDF scan (HNAS start time) or later on using the MRMT command (see below). For example:

```
HNASCMD-> DRMT MCH1 USSTAB
HNASXEQ-> DRMT RNM=MCH1 USSTAB
RMTNAME   OPERAND  TYPE
MCH1      008242B4 MCH
          USSTAB=ZZUSST1  (000D8D30) LOADED AT 08:53:29 ON 2007/04/09
```

  Note: The DRMT support described above was introduced into 240 with APAR 2400032.

- The **ECHOXEQ** command added to control how much HNASXEQ> data is echoed back to console operator.

  ECHOXEQ MAX causes 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.

  ECHOXEQ MIN causes 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. <u>ECHOXEQ MIN is the default</u>.

  ECHOXEQ NO causes the echoing of command modifiers and parameters to be inhibited.

- The **EXEC *ddname*** command now allows you to specify a DDNAME that exists in the HNAS start job which points to a file that contains a list of console commands. In the initial implementation, 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
...
```

  Note: Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

- The **EXEC *ddname*** command now stops queuing commands when an embedded EXEC command or the new END command is decoded. This will reduce the likelihood of receiving the following error message:

```
NASC054E EXEC CMDLIST1 FILE EXCEEDS QUEUED COMMAND LIMIT, EXEC COMMAND ABORTED
```

  If more than 512 bytes of command data and data length bytes are listed in the command list identified by the *ddname* argument, the NASC054E message will be generated. Prior to APAR 2400034, this would occur even if the command list contains an embedded EXEC command (which ends the current command list) listed early in the command list file. The problem occurred because the decode of the embedded EXEC command was not performed until after the entire command list was read into memory. Some users like the ability to list many commands in an EXEC command list beyond the EXEC command for use at a later time. The old logic prevented large numbers of saved commands to be listed in a command list file. The new logic will allow any number of commands to be saved after the END or embedded EXEC command.

  Note: EXEC command list END statement and suspend queuing support was introduced into 240 as Enhancement APAR 2400034.

- The **EXEC *ddname*** console command now allows multiple commands to be specified on a single record within the command list file identified by *ddname*.

  Multiple commands can be specified on a single record as follows:

  (*cmd* , ... , *cmd*) ; *comment*

  A single command can still be specified on a single record as follows:

  cmd ; *comment*

  Note: This new EXEC support was introduced into 240 as Enhancement APAR 2400064.

- The **EXEC** (***command1***,***command2***,***command3***,...) console commands can now be entered on a single line without having to invoke a ***ddname***. Prior to the 240 EXEC command only multiple non display type commands could be entered on a single line. For example, if EXEC (DNAS,DPARM) were entered, the DNAS and DPARM commands would executed consecutively.

- The **EXEC LIST *ddname*** command is now available that allows the console operator to view externally defined command lists (command lists specified via CMDLIST JCL DD statements) to determine what command sequences are associated which each command list.

- The **EXEC LIST** command has been modified to issue the new NASC052E error message (see above) if ***ddname*** is omitted.

  Note: NASC052E error message was introduced into 240 as Enhancement APAR 2400034.

- The **EXEC STOP** command is now available to terminate PAUSE command delay mode or CMDLIST execution.

- The **EXEC *ddname*** command was modified to allow CMDLIST statements of the form # *text* to be displayed as comments following the HNASCMD-> prefix. Input record that starts with the character # are not processed as commands but are simply displayed.

- The **EXEC LCLCONS|*rconname action*** command support added to allow one console to queue commands and alter queued command processing for another console.

  ```
  action = (cmd1,...,cmdn) <- Execute inline command list
           ddname          <- Execute dataset command list
           LIST ddname     <- List dataset command list
           STOP            <- Stop command list execution
  ```

  Note: The EXEC LCLCONS|*rconname action* change was introduced into 240 with Enhancement APAR 2400075.

- The **EXEC** command has been modified to decode the new NOPURGEONERROR and PURGEONERROR keywords. One of these keywords may be entered as the first value in an inline command list (**NOPURGEONERROR|PURGEONERR,*cmd*1,...,*cmd*n**) or as the first non-comment record in a *ddname* command list. For detailed usage information, please see description of the NOPURGEONERROR|PURGEONERROR options for the EXEC= operand on page D-17.

- The **HELP** command now displays common command parameters after the HELP header message. For example, when HELP ? is entered the following is displayed:

```
HNASCMD-> HELP ?
HNASXEQ-> HELP HELP
COMMAND   DESCRIPTION (* => PRIVILEGED)
          COMMON PARAMETERS FOR ALL COMMANDS: cmdname parm
                 parm=? to display help for cmdname.
                      BPM to bypass permanent modifiers for cmdname.

HELP      DISPLAY CONSOLE COMMAND HELP INFORMATION
          ENTER> HELP {SHOWALL|command}

          General notes for the HELP command:

           1) HELP with no arguments will display a one line
              description for each command.
           2) HELP command will display a verbose description for
              the given command.
           3) HELP SHOWALL will display a verbose description for
              all commands.
```

- The **HELP** command now accepts **SHOWALL** as a new keyword argument. **HELP SHOWALL** will display the complete help text for every HNAS command (verbose). HELP (with no argument) will continue to display a one-line description for each command.

- The **MMEM EOTKEY=*dd...dd* command** allows the TRIAL PERIOD EXPIRATION DATE (**EOT-DATE**) for a trial distribution to be extended using a special 16 decimal digit key provided by the ***dd...dd*** digits. An HNAS trial distribution is normally shipped with an EOTDATE that specifies when the TRIAL AUTHORIZATION will expire. A unexpired EOTDATE is required to use HNAS. In the past, the only way to extend the EOTDATE was to order and install a new refresh distribution. The new MMEM EOTKEY=***dd...dd*** command now allows the EOTDATE to be extended without the need of a new refresh distribution or without having to stop and re-start HNAS to specify the EOT-KEY=*dd*...*dd* start parameter. The EOTKEY=*dd*...*dd* start parameter is also provided so that the EOTDATE can be extended without having to issue the MMEM EOMKEY=*dd*...*dd* command each time HNAS is stopped and re-started. Please see PARM='EOTKEY=*dd*...*dd*' description on page D-15 for additional information.:

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

- The **MMEM EOMKEY=*dd...dd*** command allows the MAINTENANCE/USE ANNIVERSARY DATE (**EOMDATE**) for a permanent distribution to be extended using a special 16 decimal digit key provided by the ***dd...dd*** digits. An HNAS permanent distribution is normally shipped with an EOMDATE that specifies when the MAINTENANCE/USE license will expire. A 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 to specify the EOMKEY=*dd*...*dd* start parameter. 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 PARM='EOMKEY=*dd*...*dd*' description on page D-15 for additional information.

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

- **PARM='EOMKEY=*dd...dd*'** also allows a trial distribution to be converted to a permanent distribution dynamically using the key provided by the ***dd...dd*** digits.

  Note: EOMKEY= trial conversion logic was introduced into 240 via APAR 2400106.
-
- The **MON TAP** command NAS251*n*M monitor messages can now be filtered using the ALRMFLTR= operand so that they can be optionally routed to SYSCONS using the (FC) or (FU) suffixes.
  For example specifying ALRMFLTR=(...,NAS251*M(FU),...) on BUILD in the CDF or issuing the ALARM FILTER=(...,NAS251*M(FU),...) console command will route the NAS251*n*M monitor messages to SYSCONS as well as SYSPRINT.

- The **MON TAP** command now accepts **PKTDATA|MAXDATA|MINDATA|NODATA** as arguments to control the type of information displayed in the NAS251*n*M monitor messages.

  For PKTDATA (the default), PKT=xx...xxx data is displayed after the NAS251*n*M message header.

  For MINDATA, DTEADDR=dd...dd is displayed after the NAS251*n*M message header.

  For MAXDATA, DTEADDR=dd...dd is displayed after the NAS251*n*M message header and for the NAS2513M message, additionally the FAC= and CUD= values.

  For NODATA, only the NAS251*n*M message header is displayed.

  Note: MONTAP PKTDATA|MAXDATA|MINDATA|NODATA argument support was introduced into 240 as Enhancement APAR 2400044.

- The **MRMT LUNAME=|SVC0=|SVC4=|SVC5=*sluname*-{A|I}** command now permits manipulation of SLU state (A is for active (online) and I is for idle (offline)). This command will now propagate the VARY *sluname* {ON|OFF} command based on the SLU state provided.

  Note: *sluname*-{A|I} support was introduced into 240 as Enhancement APAR 2400028.

- The **MRMT** command has been modified to allow an asterisk to be removed from or added to the *pluname* suboperand of the GATE LUNAME= operand for a TYPE=MCH|XTP REMOTE definition statement. The presence or absence of the asterisk dictates whether a REQSESS request is presented to VTAM to solicit a BIND from the CTCP application. If an asterisk is present, HNAS will issue a REQSESS request to the application named *pluname*. If the asterisk is absent, HNAS will simply wait for the application to BIND the SLU.

  To remove, enter: MRMT {RNM=}*mchname* LUNAME=*sluname*/*pluname*

  To add, enter: MRMT {RNM=}*mchname* LUNAME=*sluname*/*pluname**

  Note that you can also use this command to change the PLU name.

  Note: The MRMT support described above was introduced into 240 with APAR 2400008.

- The **MRMT *rmtname* IPADDR=*aaa.bbb.ccc.ddd*** command has been modified to allow the IP address to be modified dynamically for TYPE=XOT REMOTEs. This ability was previously reserved for TYPE=DMY REMOTEs only. The remote must be OFFLINE in order to alter the IP address.

  Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

Note: The MRMT support described above was introduced into 240 with APAR 2400010.

- **MRMT OPTIONS=[NO]CLOSEONTAPFAILURE** command allows a customer to toggle this option for TYPE=XOT|XTP REMOTEs.

  Note: MRMT OPTIONS=[NO]CLOSEONTAPFAILURE command support was introduced into 240 as Enhancement APAR 2400055.

- **MRMT OPTIONS=[NO]CUD0SELECTSLU** command allows a customer to toggle this option for TYPE=MCH|MXT|XTP REMOTEs.

  Note: MRMT OPTIONS=[NO]CUD0SELECTSLU command support was introduced into 240 as Enhancement APAR 2400074.

- **MRMT OPTIONS=[NO]DELAYBINDRESP** command allows a customer to toggle this option for TYPE=MCH|XTP REMOTEs.

- **MRMT OPTIONS=[NO]IDTST** command allows a customer to toggle this option for TYPE=DFX REMOTEs.

  Note: MRMT OPTIONS=[NO]IDTST command support was introduced into 240 as Enhancement APAR 2400074.

- **MRMT OPTIONS=PVCRECONTMR=*secs***, **PVCSETUPTMR=*secs*** and **PVCSETUPREJ=*code*** command allows a customer to delete, change or add these options for TYPE=MCH|MXT REMOTEs.

  Note: MRMT OPTIONS=PVCRECONTMR=,PVCSETUPTMR=,PVCSETUPREJ= command support was introduced into 240 as Enhancement APAR 2400059.

- **MRMT OPTIONS=[NO]RESETINO** command allows a customer to toggle this option for TYPE=MCH|XTP REMOTEs.

  Note: MRMT OPTIONS=[NO]RESETINO command support was introduced into 240 as Enhancement APAR 2400081.

- **MRMT OPTIONS=SVCCALLTMR=*secs*** allows customer to delete, change or add this option for TYPE=MCH|MXT REMOTEs.

  Note: MRMT OPTIONS=SVCCALLTMR= command support was introduced into 240 as Enhancement APAR 2400069.

- The **MRMT** command has been modified to allow a generic SLU name to specified for a TYPE=MCH REMOTE. Syntax is **MRMT SVC0|5=*sluname+gluname/*...**.

  Note: *gluname* support was introduced into 240 as Enhancement APAR 2400084.

- The **MRMT** command has been modified to allow the tables specified for the LOGTAB=, USSTAB= and OPTIONS=NRITAB= operands to be reloaded into memory. This means that changes to tables that are made while HNAS is running can be put in to effect without having to stop and restart HNAS.

  Syntax: `MRMT rmtname LOGTAB=`*`lgtbname`*`-R`

```
MRMT rmtname USSTAB=ustbname-R
MRMT rmtname OPTIONS=NRITAB=nrtbname-R
```

When the -R follower is omitted, old logic is used. That is, the given *lgtbname*, *ustbname* or *nrtb-name* is compared against all LOGTAB=, USSTAB= or NRITAB= names, respectively, currently in use. If the name found, the given LOGTAB=, USSTAB= or NRITAB= is copied from the *found* REMOTE to the *specified* REMOTE. If the name is not found, the given LOGTAB=, USSTAB= or NRITAB= is LOADed into memory.

When the -R follower is specified, the given table 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.

Note: The MRMT 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
```

- **PARM='DBUG TCP'** start parameter allows certain TCPIP 198 ABENDs to be bypassed, replaced by new NAS2109S and NAS2110S alarm messages - HNAS execution continues. When this option is specified, the following 198 ABENDs are bypassed:

```
HALT AT LOC 80065CB4 IN NASUTIL : TCPIP INTERRUPT UNEXPECTED
HALT AT LOC 80070952 IN NASTCP  : TCPIP REPLY ID FAILURE
```

- A new **PAUSE *seconds*** command is now available that suspends or delays execution of the console command following the PAUSE commands by the seconds value. This command is useful in delaying command execution in the new CMDLISTs or in the existing CONCMDQ=list execution after initial activity has subsided.

- **PING** command now produces NAS261*n*M monitor messages that can be filtered using the ALRMFLTR= operand so that they can be optionally routed to SYSCONS using the (FC) or (FU) suffixes. For example specifying ALRMFLTR=(...,NAS261*M(FU),...) on BUILD in the CDF or issuing the ALARM FILTER=(...,NAS261*M(FU),...) console command will route the NAS261*n*M monitor messages to SYSCONS as well as SYSPRINT.

  Note: PING NAS261*n*M message support was introduced into 240 with APAR 2400022.

- The **PING (XOT)** command has been modified allowing all special ping XOT Call Request values to be coded on a TYPE=DMY REMOTE. This allows the user to issue **PING *dmyname*** so that the ping can be associated with predefined XOT Call Request packet values for router or interface level PINGing.

  The syntax is: `[RNM=`*`dmyname`*`] PING [RNM=]`*`dmyname`*` [`*`parms`*`]`

  You can set a permanent TYPE=DMY REMOTE with the left side RNM= modifier and override it with a right side *dmyname* (the RNM= keyword is optional on the right). The *parms* operand are existing PING parameters that can be entered to override values on the dummy. For example, you can predefine an IPADDR on the dummy and override when the PING command is entered via PING *dmyname* 1.2.3.4 (for example). Note also that you can use MRMT to modify any supported dummy operand. The TYPE=DMY REMOTE now accepts PROTOCOL=XOT|XTP so you don't have to specify a protocol on the PING command although you can override it with PING *dmyname* XTP (for example). This operand can also be changed with the MRMT command.

- The **PRNT** console command now supports the **{PRTSWLST|SWITCH}=(?|*|*list*)** parameter which is used to display, clear, switch and/or update the actions and active *ddname*i|DYNAMIC list as initially specified by the PRTSWLST= operand on the BUILD definition statement. Please refer to the PRNT command description located in the Console SubSystem documentation for more information.

- The **PRNT** console command now supports the **NEXTPRSW** parameter, or equivalently the **SWITCHNOW** parameter, which is used to force a switch to the next *ddname*i|DYNAMIC SYSPRINT log in the BUILD PRTSWLST= operand. Please refer to the PRNT command description located in the Console SubSystem documentation for more information.

- The **PRNT** console command now supports the **RSMEPRSW** parameter which is used to restart SYSPRINT switching in the BUILD PRTSWLST= operand after all datasets have been used and the STOP action is in effect. Please refer to the PRNT command description located in the Console SubSystem documentation for more information.

- The **PRNT** console command now supports the **TIMESTAMP=(*time-format*,+D|-D)** parameter which is used to updates TIMESTAMP time and date controls.

- The **PRNT ON|OFF ... DATE ON|OFF ...** console command now generates a NAS01*nni* alarm message when a state change is effected by the command.

  Note: The new PRNT alarm messages were added to 240 as part of Enhancement APAR 2400017.

- The **QUIT** command now accepts **E** a new keyword follower. **QE {*ddname*|(*cmd*1,...,*cmd*n)}** will cause the command list identified by *ddname* or supplied by the specified (*cmd*1,...,*cmd*n) list to be executed before HNAS is shutdown. In addition, QUIT help has been enhanced.

- The **QUIT** shutdown process now executes **DNAS** and **ALARM LOG=?** console commands unconditionally when any QUIT command is entered. If QE ddname entered, the target list is appended to the forced commands.

- **SCHEDULE {*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)|LIST|SUSP|RSME|PRG}** console command added to allow scheduled console commands to be added, deleted or displayed.

  Syntax: SCHEDULE {*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)|LIST|SUSP|RSME|PURGE}

  (*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*) is an *inline* schedule list. An *inline* schedule list is added to the end of an existing schedule list if one is active.

  ddname identifies a file that contains a list of times and commands. A *ddname* schedule list is added to the end of an existing schedule list if one is active. A *ddname* schedule list can contain the following records:

  Comments are allowed and start with an asterisk (*) or semi-colon (;) in record column 1. Comments can also appear on a command line but must start with a semi-colon after the command. A single time and command can be specified on a single record as follows:

  hh:*mm*:*ss* , *cmd* ; *comment*

  Multiple times and commands can be specified on a single record as follows:

  (*hh*:*mm*:*ss* , *cmd* , ... , *hh*:*mm*:*ss* , *cmd*) ; *comment*

  LIST [*ddname*] is used to display a schedule list. If *ddname* is not specified, the active (current) schedule list is displayed. If *ddname* is specified, the identified schedule list is displayed but the active schedule list remains unaffected.

  SUSP causes the active schedule list to be suspended so that no commands will be scheduled. The list itself is maintained.

  RSME causes the active schedule list that was suspended to be resumed so that commands will be scheduled again.

  PRG causes the current schedule list to be cleared. No commands are left to be scheduled.

  Note: Please refer to the **Enhancements - Expanded Content** heading at the end of this section for additional information regarding this new or enhanced feature.

  Note: The new SCHEDULE command support was introduced into 240 as Enhancement APAR 2400064.

- The **SCHEDULE LCLCONS|*rconname action*** command support added to allow one console to schedule commands and alter scheduled command processing for another console.

```
action = (time1,cmd1,..) <- Append inline schedule list to active schedule list
         ddname          <- Append dataset schedule list to active schedule list
         LIST ddname     <- List dataset schedule list
         LIST            <- List active schedule list
         SUSP            <- Suspend active schedule list
         RSME            <- Resume active schedule list
         PRG             <- Purge active schedule list
```

Note: The SCHEDULE LCLCONS|*rconname action* change was introduced into 240 with Enhancement APAR 2400075.

- The **SHOW CMSG ON|OFF** command has been added to manipulate the **SHOWCMSG** start parameter.

  Note: The SHOW CMSG ON|OFF console command support was introduced into 240 with APAR 2400036.

- The **SHOW ON|OFF|ERR ... MORE|LESS ...** console command now generates a NAS02*nni* alarm message when a state change is effected by the command.

  Note: The new SHOW alarm messages were added to 240 as part of Enhancement APAR 2400017.

- New **SHOWCMSG** parameter 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:

  <u>NAS3799I Without Compression (SHOWCMSG OFF)</u>:

```
0         1         2         3         4         5         6
0123456789012345678901234567890123456789012345678901234567890123456789
|                                                                    |
NAS3799I LU MCH1PVC  ENDING   SESSION ON MCH MCH1     RMT  CAUSE/DIAG=
000/130 (00/82) DIAGX=0000
|                         |
01234567890123456789012345
7         8         9
```

  <u>NAS3799I With Compression (SHOWCMSG ON)</u>:

```
0         1         2         3         4         5         6
0123456789012345678901234567890123456789012345678901234567890123456789
|                                                                    |
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.

- **SMSG LCLCONS|*rconname* '*text*'** command support added to allow the target console to be identified by PCE name.

  Note: The SMSG LCLCONS|*rconname* '*text*' change was introduced into 240 with Enhancement APAR 2400075.

- **SNAP ALL|*trgtlist*** command support added to allow specific HNAS storage areas to be dumped in the SYSPRINT file.

```
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
```

  Note: The new SNAP command support was introduced into 240 as Enhancement APAR 2400080.

- **TCPIP** external interrupt table entries extended to allow logging of additional diagnostic information making HNAS TCPIP tracing less of a requirement for problem diagnosis. New values are TCPIP target/class (command/socket ID), RC/ERNO, start/end times.

- The **TRCCONS LCLCONS|*rconname* {ON|OFF}** command support added to allow the target console to be identified by PCE name.

  Note: The TRCCONS LCLCONS|*rconname* {ON|OFF} change was introduced into 240 with Enhancement APAR 2400075.

- The **TRCMCH {ICR|ICRF|OCR|ICLR|OCLR}** followers are now allowed as a start parameters to enable *global* Call and Clear request trace logging. **TRCMCH GBL {ICR|ICRF|OCR|ICLR|OCLR}** are the equivalent console commands.

  The **TRCMCH** console 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 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. RNM= or ID= must be set if LCL is specified. *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).

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. RNM= and ID= are ignored if GBL is specified. *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).

If you want ICRF, ICR, OCR, ICLR and/or OCLR to have *global* significance, you should enter GBL immediately after the TRCMCH command (e.g., TRCMCH GBL ICR ... OCLR). GBL is assumed if RNM= and ID= are not set (both are null). LCL is assumed if GBL is omitted and RNM= or ID= are set.

- The **TRCSUBR** start parameter and console command now allow an event list to be provided in addition to the normal ON|OFF arguments. The syntax is **TRCSUBR {ON|OFF} {*eventlist*}** where ***eventlist***={CONS|MCH|NETV|PCE|TCP|VTAM}. ***eventlist*** 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.

  History: When TRCSUBR is in effect, every subroutine within HNAS logs a number of trace entries. Some are very useful but others are simply unwanted 'noise'. 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 now 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. HELP for the TRCSUBR command will now display the following:

```
TRCSUBR  *SUBROUTINE CALL TRACE CONTROL

          ENTER> TRCSUBR {ON|OFF} {eventlist}

          eventlist = {ALLEVENTS|NOEVENTS} or
                      {CONS|NETV|TCP|VTAM|MCH|PCE}

          General notes for the TRCSUBR command:

          1) Enter ON to initiate global subroutine call tracing.
```

```
2) Enter OFF to terminate global subroutine call tracing.
3) Enter ALLEVENTS to set the subroutine call trace event
   filter to all events.
4) Enter NOEVENTS to reset the subroutine call trace event
   filter to no events.
5) Enter CONS, NETV, TCP, VTAM, MCH and/or PCE to set the
   subroutine call trace filter to specific events.  You may
   enter NOEVENTS followed by a list of events to first clear
   then repopulate the subroutine call trace event filter.
6) ON is assumed if TRCSUBR is entered with no argument.
7) If NOEVENTS is in effect when ON is set, ALLEVENTS is
   forced.
8) Tracing requires additional CPU cycles.
```

Note: TRCSUBR ***eventlist*** logic was introduced into 240 via APAR 2400108.

- Enhancements to **TRCTRAP ALRMLIST=(NAS*nnnni*/*data*/*data-offset*,...)** now provides the ability to not only restrict a trap 'hit' to a specific message ID but also to data within the message. When the *data* suboperand is provided, the trap is not sprung when a message ID match occurs unless the message also contains the text *data* you provide (*data* is not enclosed in quotes and no spaces are permitted). The *data* is used to further restrict the message search to specific data content. For example, the DIAG= value in a NAS7715W message. The *data-offset* value, which can range from 1 to 128, identifies where in the message the *data* can be found. The *data-offset* allows the *data* search to start at a specific offset within the message which reduces CPU utilization when looking for a match. The *data-offset* value is relative to the first character of a message, that is, the beginning of the message ID (NAS*nnnni*). If no *data-offset* is given, the *data* search starts at the beginning of the message. If a *data-offset* value plus the *data* length exceeds the message length, a 'hit' is assumed for the message ID only.

- The ***data*** suboperand of the **TRCTRAP ALRMLIST=(NAS*nnnni*/'*data*'/*data-offset*,...)** command will now accept a quoted string. This is required so that a forward slash (/) wihin the data is treated as part of the data and not as a suboperand delimiter.

  History: If ***data*** contains a forward slash (/), the slash will be treated as a delimiter making the data that follows it being treated as the ***data-offset*** suboperand. This can cause the entire ALRMLIST value to be ignored and an error condition to be set. For example, when the following TRCTRAP console command is entered:

  TRCTRAP ALRMLIST=(NAS3799I/DIAG=000/195)

  the following error message is issued and the request is rejected:

  NASC532E PARAMETER DATA INVALID: IAG=000/195)...,
  TRCTRAP COMMAND ABORTED

  This is occurring because the ***data*** value DIAG=000/195 contains a forward slash making the 195 that follows the slash treated as a ***data-offset*** value. Since 195 is too large for a ***data-offset*** value, the command is rejected.

  The TRCTRAP CDF operand processor and the TRCTRAP console command processor have been modified to accept the ***data*** 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.

Note1: If ***data*** does not contain spaces or a forward slash, it does not have to be specified within quotes, but can be if you wish. ABCDEF and 'ABCDEF' are treated the same.

Note2: ***data*** may not include embedded quotes. For example, ABC'DEF or ABC"DEF or 'ABC'DEF' or 'ABC"DEF' are not permitted.

Note: TRCTRAP quoted '***data***' logic was introduced into 240 via APAR 2400108.

- The **TRCTRAP TRAPACTION=(EXEC=*ddname*)** option now supports command list execution upon trap action filter condition. This allow for automation of various console command actions for debugging without operator intervention.

- The **TRC*type*** console commands have been modified to accept multiple arguments for the same command invocation. For example, TRCMCH ICR ICRF OCR ... OCLR will now be valid. In the past, you were required to enter each option via a separate command. For example TRCMCH ICR ... TRCMCH OCLR. This syntax is still valid but the multiple argument syntax is also accepted.

- The **VARY *mchname* {OFF|ON}** command has been added to the VARY command. This permits or restricts MCH resources to be used for inbound or outbound calls. An inactive MCH will be skipped when processing the RTEIN= operand.

- The **VARY *sluname* {OFF|ON}** command has been added to the VARY command. This permits or restricts SLU resources to be used for inbound or outbound calls.

- The **VARY *sluname* FORCE** command has been modified to close the TCP/IP socket associated with the SLU even if the SLU itself is inactive (normally only PVCs can be in this state). Prior to this change, the command would have been rejected if the SLU was inactive with the following error message:

```
NASC015E FUNCTION ALREADY INACTIVE, VARY COMMAND ABORTED
```

Note that this message can still be issue for VARY *sluname* FORCE if the SLU does not have an active TCP/IP socket connection or for VARY *sluname* OFF is the SLU is inactive.

This new action for VARY *sluname* FORCE eliminates the need to issue the DLU *sluname* command to obtain the *rpid* value for the SLU followed by the VARY ID=*rpid* FORCE command to close the associated TCP/IP socket.

Note: The new VARY *sluname* FORCE change was introduced into 240 by Enhancement APAR 2400098.

Enhancements - Expanded Content:

- The **EXEC *ddname*** command allows you to specify a DDNAME that exists in the HNAS start job which points a file that contains a list of console commands.

  In the initial implementation, 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
```

  These command list files can be executed using the new EXEC console command as follows:

  EXEC CMDLIST1
  EXEC CMDLIST2
  EXEC CMDLIST3

  If a ddname is specified that does not exist in the start JCL, the following message is issued:

  NASC051E EXEC badddnm INVALID, CANNOT BE OPENED

  When a valid command list ddname is specified and it has been read successfully, the following message is generated:

  NASC055I EXEC CMDLIST1 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 and 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. Note that in the initial development, 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.

```
//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
```

  EXEC CMDLISTS

  Again, all DDNAMEs and the members they reference are totally arbitrary. The only requirement is that the ddname you provide for the new EXEC command MUST exist in the HNAS start JCL.

  Note that if DISP=SHR is specified for all command list ddnames above, it will allow you to modify the command lists when 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.

Note that the command list file can contain comments that start with an asterisk (*) or semi-colon (;) in column 1. Note also that each individual record in a command list file can contain a comment that follows the command text if the first character is a semi-colon (just like the CDF parsing). Command text is assumed to start with the first non-blank character on a line and end with the last non-blank character on a line (prior to the comment start ; if it is present). 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
  DNAS                        ; FIRST QUEUED COMMAND
  DPARM EXEC                  ; SECOND QUEUED COMMAND
```

This file would be processed as though CONCMDQ=(DNAS,DPARM) were specified.

Currently the BUILD CONCMDQ= operand allows a maximum of 512 bytes for queued commands. This is also true for commands that are provided by the EXEC ddname command. Note that the 512 byte maximum includes a length byte for each queued command. For example: If DNAS and TRC-MCR ICR were provided in the CONCMDQ= operand or in a command list, the total number of bytes that will be saved is 5 for DNAS (length byte = 4) and 11 for TRCMCH ICR (length byte = 10). If you provide enough date in a command list that exceeds the 512 byte limit, the following messages will be issued:

NASC054W EXEC XXXXXXXX FILE EXCEEDS QUEUED COMMAND LIMIT,
COMMAND LIST IGNORED

- The **MRMT *rmtname* IPADDR=*aaa.bbb.ccc.ddd*** command has been modified to allow the IP address to be modified dynamically for TYPE=XOT REMOTEs. This ability was previously reserved for TYPE=DMY REMOTEs only.

  In order to allow this change, the target TYPE=XOT REMOTE must be in the offline state. This can be done in the CDF by specifying INIT=IDLE or during execution using the VARY FORCE command. For example, assume that the TYPE=XOT REMOTE named R1CNIN was configured with IPADDR=10.117.56.100 and INIT=ACTIVE. To change the IP address from its current value of 10.117.56.100 to 10.117.56.201, for example, the following command sequence would have to be entered:

```
RNM=R1CNIN                    <- set remembered RMT name
VARY FORCE NOPR               <- vary R1CNIN offline
MRMT IPADDR=10.117.56.201     <- change IP address
VARY ON NOPR                  <- vary R1CNIN back online
```

Notes: 1) The commands above could have been entered with a temporary REMOTE name as a right side modifier, e.g., VARY R1CNIN FORCE NOPR

2) If the MRMT command above is issued while the REMOTE is online, the command will be rejected and the following message issued:

```
NASC022W RESOURCE ONLINE,  MUST BE OFFLINE (USE VARY)
```

3) If the MRMT command attempts to change a dynamic IP address (IPADDR=DYNAMIC), the command will be rejected and the following message issued:

```
NASC312E RNM=rmtname CONFIG INVALID, REQUIRED
```

- **NAS1391I** messages now issued to display MXT operand overrides. When a user associates an MXT with a PVC= or SVC0/3/5= operand entry, he is asking HNAS to override operands on the root MCH with those from the MXT. In some cases, this can lead to confusion if one does not remember to review the HNAS documentation to see which operands can be overridden based on the associated resource type.

  The configuration logic has been modified to display, for each SLU or SPU defined in the PVC= or SVC0/3/5= operands with an associated MXT (or SVC), those MXT operands that will override operands of the same name on the root MCH. Not all MXT operands are valid overrides for all SLUs or SPUs. For example, IDLETO= is a valid override when specified for an MXT that is associated with a SVC0/5= SLU but is not used for a PVC SLU.

  MXT override messages are listed at the end of the CDF scan in NAS1391I messages. Only those MXT operands that are specified and, of these, only those that are valid as overrides for an SLU or SPU are displayed. If an MXT operand is not listed, it is because it was either omitted from the MXT (an hence cannot override the MCH operand) or, if specified, is not valid as an override for the SLU or SPU resource. Consider the following configuration example:

```
MCH1     REMOTE TYPE=MCH
               IDLETO=2
               PAD=INTEG
               PADPARM=(1/1,
                        2/1,
                        13/7,
                        21/0,
                        110/0)
               PVC=(3,
                    MCH1P001/5/255/01/SERIAL0-1/R2CNOT1/MXT0,
                    MCH1P002/5/255/02/SERIAL0-1//MXT2,
                    MCH1P003/5/255/03/SERIAL0-1)
               SVC0=(2,
                     MCH10011/1234-X1234-<MXT1>T01/MXT0
                     MCH10012/1234-<MXT0>-X1234T1)
               SVC5=(2,
```

```
                    MCH15001/1234-X1234-<MXT1>T01/MXT0
                    MCH15002/1234-<MXT0>-X1234T1)
              SVC3=(1,Q1990100//MXT0)
              :
MXT0     REMOTE TYPE=MXT
              CUD=(01111111)
              DCEADDR=1111
              DTEADDR=9999
              FAC=(420808
                   430404)
              IDLETO=15
              PADPARM=(1/1,
                       2/1,
                       3/2,
                       4/0,
                       5/2,
                       7/2,
                       12/1,
                       13/7,
                       21/0,
                       110/0)
MXT1     REMOTE TYPE=MXT
              CUD=(01222222)
              DCEADDR=2222
              FAC=(0101420A0A430202)
              PADPARM=(1/1,
                       2/1,
                       7/2,
                       12/1,
                       13/7,
                       21/0,
                       110/0)
MXT2     REMOTE TYPE=MXT
              CUD=(01222222)
              DCEADDR=2222
Q1990100 REMOTE TYPE=SPU
              LUNAME=(,
              Q1990102/201/201/1/PVCMXT,
              Q1990103)
              DTEADDR=20361234
              DCEADDR=20364321
              FAC=420707430707
              CUD=C3000000
              IDBLK=199
              IDNUM=00001
              :
```

Based on this configuration, the following messages are generated at the end of the CDF scan.

```
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH1P001 (DEFINED ON MCH MCH1
NAS1391I AS PVC ENTRY 000) COME FROM MXT MXT0     RATHER THAN THE MCH
NAS1391I --> FAC=420808430404
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH1P002 (DEFINED ON MCH MCH1
NAS1391I AS PVC ENTRY 001) COME FROM MXT MXT2     RATHER THAN THE MCH
```

```
NAS1391I --> NO OVERRIDES
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH10011 (DEFINED ON MCH MCH1
NAS1391I AS SVC0 ENTRY 0000) COME FROM MXT MXT0     RATHER THAN THE MCH
NAS1391I WHEN CONNECTION IS NOT ESTABLISHED VIA A <RMTNAME>
NAS1391I --> CUD=01111111
NAS1391I --> DCEADDR=1111
NAS1391I --> DTEADDR=9999
NAS1391I --> FAC=420808430404
NAS1391I --> IDLETO=00015
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH10011 (DEFINED ON MCH MCH1
NAS1391I AS SVC0 ENTRY 0000) COME FROM MXT MXT1     RATHER THAN THE MCH
NAS1391I WHEN CONNECTION IS ESTABLISHED VIA <MXT1>
NAS1391I --> CUD=01222222
NAS1391I --> DCEADDR=2222
NAS1391I --> FAC=0101420A0A430202
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH10012 (DEFINED ON MCH MCH1
NAS1391I AS SVC0 ENTRY 0001) COME FROM MXT MXT0     RATHER THAN THE MCH
NAS1391I WHEN CONNECTION IS ESTABLISHED VIA <MXT0>
NAS1391I --> CUD=01111111
NAS1391I --> DCEADDR=1111
NAS1391I --> DTEADDR=9999
NAS1391I --> FAC=420808430404
NAS1391I --> IDLETO=00015
NAS1391I THE FOLLOWING OPERANDS FOR SPU Q1990100 (DEFINED ON MCH MCH1
NAS1391I AS SVC3 ENTRY 0000) COME FROM MXT MXT0     RATHER THAN THE MCH
NAS1391I WHEN THEY ARE OMITTED FOR THE SPU
NAS1391I --> CUD=01111111
NAS1391I --> DCEADDR=1111
NAS1391I --> DTEADDR=9999
NAS1391I --> FAC=420808430404
NAS1391I --> IDLETO=00015
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH15001 (DEFINED ON MCH MCH1
NAS1391I AS SVC5 ENTRY 0000) COME FROM MXT MXT0     RATHER THAN THE MCH
NAS1391I WHEN CONNECTION IS NOT ESTABLISHED VIA A <RMTNAME>
NAS1391I --> CUD=01111111
NAS1391I --> DCEADDR=1111
NAS1391I --> DTEADDR=9999
NAS1391I --> FAC=420808430404
NAS1391I --> IDLETO=00015
NAS1391I --> PADPARM=001/001 002/001 003/002 004/000 005/002 007/002 ..
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH15001 (DEFINED ON MCH MCH1
NAS1391I AS SVC5 ENTRY 0000) COME FROM MXT MXT1     RATHER THAN THE MCH
NAS1391I WHEN CONNECTION IS ESTABLISHED VIA <MXT1>
NAS1391I --> CUD=01222222
NAS1391I --> DCEADDR=2222
NAS1391I --> FAC=0101420A0A430202
NAS1391I --> PADPARM=001/001 002/001 007/002 012/001 013/007 021/000 ...
NAS1391I THE FOLLOWING OPERANDS FOR SLU MCH15002 (DEFINED ON MCH MCH1
NAS1391I AS SVC5 ENTRY 0001) COME FROM MXT MXT0     RATHER THAN THE MCH
NAS1391I WHEN CONNECTION IS ESTABLISHED VIA <MXT0>
NAS1391I --> CUD=01111111
NAS1391I --> DCEADDR=1111
NAS1391I --> DTEADDR=9999
NAS1391I --> FAC=420808430404
```

```
NAS1391I --> IDLETO=00015
NAS1391I --> PADPARM=001/001 002/001 003/002 004/000 005/002 007/002 ...
```

General Notes:

1) The NAS1391I message for SLU MCH1P001 above only displays the FAC= operand for MXT0 even though other operands were specified. This is because only the FAC= operand is used as a PVC SLU override operand.

2) The NAS1391I message for SLU MCH1P002 shows 'NO OVERRIDES' because the FAC= operand for was omitted for MXT2. Note that no NAS1391I message is issued for SLU MCH1P003 since no MXT was associated with this SLU.

3) The NAS1391I messages for the SVC3= operand above list MXT overrides when operands are omitted from the SPU. The order is MXT first, SPU then MCH. This only applies when SPU connections are established via the MCH and not by IDBLK/IDNUM matching which is MCH independent.

4) The SVC0= and SVC5= operands above are identical except for the SLU names. Note that the NAS1391I messages for the SVC0= and SVC5= operands are the same except that the PADPARM= operand is listed for SVC5= SLUs (when specified on the MXT) while it is not for SVC0 SLUs. This is because only SVC5= SLUs use the PADPARM= operand.

5) The NAS1391I messages are generated only when an SLU or SPU has an associated MXT as provided in the PVC= or SVC0/3/5= operand. If you wish to withhold the NAS1391I messages from the SYSPRINT log, specify ALRMFLTR=(...,NAS1391I(P),...) on the BUILD definition statement.

6) If you plan on filtering NAS1391I messages from SYSPRINT, we highly recommend that you first run a FASTRUN pass without filtering so that these important messages are at least displayed once for your perusal.

Note: NAS1391I message support was introduced into 240 as Enhancement APAR 2400018.

- **OPTIONS=(TIMESTAMP=*format*-{+D|-D}** was added to the BUILD definition statement to enhance SYSPRINT timestamp resolution.

  Specify TIMESTAMP=HH:MM:SS[.T[H[M[I]]]] to set the HNAS SYSPRINT log timestamp resolution to seconds, tenths of seconds (T), hundredths of seconds (H), thousandths of seconds (M) or tenthousandths 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 PRNTDATE ON).

- **PARSE** definition statement has been added to allow customers to specify their own parsing characters that will override established HNAS parsing characters for the CDF scan. The PARSE definition statement must be the first definition statement in the CDF and must precede any CDF records that utilize the special parsing characters that it defines. After the PARSE definition statement is processed, the override parsing characters will go into effect turning the established parsing characters into non-parsing characters. They will simply become data characters.

The following parsing override characters are currently provided:

- Comment start character

  The semi-colon (;) is normally used to indicate the start of a comment that follows configuration data on a CDF record. This character can also occur in record column 1 when the entire record is to be treated as a comment, however, an asterisk (*) in record column 1 is normally used for this purpose and cannot be overridden. To override the semi-colon as the comment start character, specify **OPTIONS=(...,COMCHAR=*comchar*,...)** on the PARSE definition statement.

- Continuation start character

  The plus-sign (+) is normally used to indicate that data on the current CDF record is to be continued on the next CDF record. To override the plus-sign as the continuation start character, specify **OPTIONS=(...,CONCHAR=*conchar*,...)** on the PARSE definition statement.

- Suboperand separator character

  The forward-slash (/) is normally used to separate suboperands for an operand entry that accepts multiple suboperands. For example, SVC0=(...,*sluname*/*dteaddr*/*mxtname*,...). To override the forward-slash as the suboperand separator character, specify **OPTIONS=(...,SEPCHAR=*sepchar*,...)** on the PARSE definition statement.

- Forced space character

  The blank ( ) is normally used to provide spacing. When CDF records are continued using the plus-sign (+) or the OPTIONS=CONCHAR=*comchar* override character, the CDF parser removes all blanks between the last non-blank character on a record and the continue character. If you want HNAS to leave spaces between the last non-blank character and the continue character on a record, you can specify a forced space character that you can use to provide a place holder for blanks that are not to be deleted. To supply a forced space character, specify **OPTIONS=(...,FSPCHAR=*fspchar*,...)** on the PARSE definition statement. For example, if OPTIONS=FSPCHAR=~ is specified, you can prevent blanks from being deleted on a continuation record as follows:

  ```
  BUILD CONCMDQ=('TRCMCH~~~+
                   ~~ICR')
  ```

  This would be processed as though CONCMDQ=('TRCMCH   ICR') were entered.

  comchar, ***conchar***, ***sepchar*** and ***fspchar*** in the above descriptions is a single EBCDIC character without framing quotes ('').

- **PRTSWLST=({LOOP|STOP},SWITCHAFTERINIT,SWITCHAT*time*,**
  {*ddname*1|DYNAMIC=*outclass*},...,{*ddname*n|DYNAMIC=*outclass*})
  was added to the BUILD definition statement to provide 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. The DDNAMEs you specify or request dynamically are used sequentially. The default SYSPRINT file is always used initially (ddname=SYSPRINT).

  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 LOOP 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=(LOOP,SWITCHAFTERINIT,SWITCHATMIDNIGHT,DYNAMIC).

The dynamic datasets are allocated with a DDNAME of DYN#*xxxx* where *xxxx*=0001 for the first dynamically allocated dataset and 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: PRTSWLST=DYNAMIC support was introduced into 240 as Enhancement APAR 2400001.

LOOP, STOP, SWITCHAFTERINIT, SWITCHAT*hh*00, SWITCHAT6AM, SWITCHATMIDDAY, SWITCHAT6PM and SWITCHATMIDNIGHT are reserved keywords that can be specified anywhere in the PRTSWLST= operand list. They are NOT treated as DDNAMEs.

If LOOP is specified as a PRTSWLST= suboperand, DDNAMEs in the list are 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 *ddname*i can be SYSPRINT in which case it will be reused.

If STOP is specified as a PRTSWLST= suboperand, SYSPRINT logging stops when *ddname*n becomes full.

If SWITCHAFTERINIT is specified as a PRTSWLST= suboperand, logging in the default SYSPRINT dataset stops and logging in the *ddname*1 dataset starts after the NAS0001I INITIALIZATION COMPLETE message is issued. This option was added so that customers can send us an abbreviated HNAS log file if we need to look at the CDF scan and DNAS console command output.

If SWITCHAT*hh*00 is specified (military time) as a PRTSWLST= suboperand, logging in the current PRTSWLST= dataset stops and logging in the next PRTSWLST= *ddname*i dataset starts at the hour specified by *hh* every day. Note that *hh* can be 01 to 24|00 (00 is treated the same as 24).

If SWITCHAT6AM (=SWITCHAT0600) is specified as a PRTSWLST= suboperand, logging in the current PRTSWLST= dataset stops and logging in the next PRTSWLST= *ddname*i dataset starts at 6AM every day.

If SWITCHATMIDDAY (=SWITCHAT1200) is specified as a PRTSWLST= suboperand, logging in the current PRTSWLST= dataset stops and logging in the next PRTSWLST= *ddname*i dataset starts at 12PM every day.

If SWITCHAT6PM (=SWITCHAT1800) is specified as a PRTSWLST= suboperand, logging in the current PRTSWLST= dataset stops and logging in the next PRTSWLST= *ddname*i dataset starts at 6PM every day.

If SWITCHATMIDNIGHT (=SWITCHAT2400) is specified as a PRTSWLST= suboperand, logging in the current PRTSWLST= dataset stops and logging in the next PRTSWLST= *ddname*i dataset starts 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 actions is specified.

For example: `PRTSWLST=(LOOP,SWITCHATMIDNIGHT,SYSPRNT1,SYSPRNT2,SYSPRNT3).`

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*i 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 RSMEPRSW command.

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.

- **SCHEDULE {*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)|LIST|SUSP|RSME|PRG}** console command added to allow scheduled console commands to be added, deleted or displayed.

  Syntax: SCHEDULE {*ddname*|(*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*)|LIST|SUSP|RSME|PURGE}

  (*hh*:*mm*:*ss*,*cmd*,...,*hh*:*mm*:*ss*,*cmd*) is an *inline* schedule list. An *inline* schedule list is added to the end of an existing schedule list if one is active.

  ddname identifies a file that contains a list of times and commands. A *ddname* schedule list is added to the end of an existing schedule list if one is active. A *ddname* schedule list can contain the following records:

  Comments are allowed and start with an asterisk (*) or semi-colon (;) in record column 1. Comments can also appear on a command line but must start with a semi-colon after the command. A single time and command can be specified on a single record as follows:

  hh:*mm*:*ss* , *cmd* ; *comment*

  Multiple times and commands can be specified on a single record as follows:

  (*hh*:*mm*:*ss* , *cmd* , ... , *hh*:*mm*:*ss* , *cmd*) ; *comment*

  LIST [*ddname*] is used to display a schedule list. If *ddname* is not specified, the active (current) schedule list is displayed. If *ddname* is specified, the identified schedule list is displayed but the active schedule list remains unaffected.

  SUSP causes the active schedule list to be suspended so that no commands will be scheduled. The list itself is maintained.

  RSME causes the active schedule list that was suspended to be resumed so that commands will be scheduled again.

  PRG causes the current schedule list to be cleared. No commands are left to be scheduled.

  <u>General Notes</u>:

1) In order to provide symmetry between SCHEDULE and EXEC command processing, the EXEC command was also modified to allow multiple commands to be specified on a single record within a command list file as follows:

(*cmd* , ... , *cmd*) ; *comment*

A single command can still be specified on a single record as follows:

cmd , *comment*

2) For both the EXEC and SCHEDULE commands, 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.

3) For both the EXEC and SCHEDULE commands, leading and trailing blanks are removed from each *cmd* (and time for the SCHEDULE command) before the values are saved.

4) For both the EXEC and SCHEDULE commands, an END command will signal the end of data in a file. In addition, for EXEC, an embedded EXEC command and for SCHEDULE, an embedded SCHEDULE command will also signal the end of data in a file. Any records that follow these statements will be ignored.

5) For both the EXEC and SCHEDULE commands, the CONCMDQ queue is used to save 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.

<u>WARNING</u>: 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.

6) Multiple commands can be scheduled for execution at the same time. For example:

(12:00:00,ALARM LOG=?,12:00:00,DNAS)

<u>WARNING</u>: Because scheduled commands are enqueued to CONCMDQ 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)

7) 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.

<u>Examples</u>:

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.

You can specify a wildcard for any time except the low order second digit (*hh*:*mm*:*s**) because a scheduled command could be executed every second which could impact HNAS performance.

8) For the SCHEDULE command when an *inline* list or *ddname* list is given, entries are added to the end of the active SCHEDULE list in effect. If the active schedule list contains another SCHEDULE command, the specified schedule list will replace the active (RESIDENT) list. This allows you to chain schedule lists (see Note 9 examples).

9) For the EXEC command, an embedded EXEC command allows you to chain command lists. For the SCHEDULE command, an embedded SCHEDULE command allows you to chain schedule lists.

For example, if you want to run a different schedule list each day, you could structure them as follows (assume HNAS is started on Monday):

```
BUILD 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 will end with a SCHEDULE command that points at the next file which will take effect at midnight.

The DD4MON file will end with the following statement: (00:00:00,SCHEDULE DD4TUE)

The DD4TUE file will end with the following statement: (00:00:00,SCHEDULE DD4WED)

:

The DD4SUN file will end 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 inactive 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 HNAS is started during the day):

```
BUILD SCHEDULE=DD4DAY

//DD4DAY DD DSN=hlq.HNASMAC(DD4DAY),DISP=SHR
//DD4NIT DD DSN=hlq.HNASMAC(DD4NIT),DISP=SHR
```

As before, each schedule list file will end with a SCHEDULE command that points at the other file which will take effect at the specified time.

The DD4DAY file will end with the following statement: (00:00:00,SCHEDULE DD4NIT)

The DD4NIT file will end with the following statement: (06:00:00,SCHEDULE DD4DAY)

10) When a scheduled command is executed, the normal console command echo prefix is changed from HNASCMD-> to SCHDCMD-> so that scheduled commands can be identified in the SYSPRINT log file.

11) When SCHEDULE=schdlist is decoded during HNAS activation, scheduled commands are suspended until after the **NAS0001I INITIALIZATION COMPLETE** message is issued and after first CONCMDQ=cmdlist processing.

Note: The new SCHEDULE command support was introduced into 240 as Enhancement APAR 2400064.

- **SVC0|5=**{...(*sluname*/{X*idnum*|*dteaddr*|**<*rmtname*>**}...} permits an MXT|SVC per *dteaddr* and a *dteaddr* via *rmtname*. New **REMOTE TYPE=SVC** was added to provide a smaller MXT footprint without PADPARM=*value* support.

This enhancement is provided so that customers can associate a separate TYPE=MXT REMOTE or the new TYPE=SVC REMOTE with each callout and callin DTE address when multiple DTE addresses are specified for an SVC0= or SVC5= operand entry. Prior to this enhancement, an LLC0 callout resource with multiple destination addresses was specified as follows:

```
MCH1 REMOTE TYPE=MCH|XTP
             SVC0=(n,sluname/dteaddr1-dteaddr2-dteaddr3{O|T}[applid]
                            /mxtname/cud,...
```

When the SLU is bound by a PLU, a Call Request packet is generated using *dteaddr*1 as the *called* DTE address. If this call fails (Clear received or no response to Call Request packet), a new call is then placed to *dteaddr*2 and, if necessary, to *dteaddr*3.

When a *dteaddr*i value is specified, the MXT identified by *mxtname* above becomes the TYPE=MXT REMOTE that is associated with the SLU for the entire LU/VC session. The *dteaddr*i value is used as the *called* DTE address while the named MXT supplies the *calling* DTE address (DCEADDR= operand), facilities data (FAC= operand), call user data (CUD= operand) and for PAD calls, PAD parameters (PADPARM= operand). When multiple *dteaddr*i values are coded, all outbound Call Request packets will carry the same *calling* DTE address, facilities data and call user data because there is only one MXT pointer.

HNAS has been modified to allow the name of an MXT (short for a TYPE=MXT REMOTE) or the new SVC (short for a TYPE=SVC REMOTE) to be specified in place of the *dteaddr*i suboperands.

When an MXT or SVC is specified as a *rmtname*i in place of a *dteaddr*i value, it becomes the REMOTE that is associated with the SLU for the duration of the LU/VC session. In this case, it will be the REMOTE that provides the *called* DTE address (DTEADDR= operand), *calling* DTE address (DCEADDR= operand), facilities data (FAC= operand) and call user data (CUD= operand). These four operands provide all the information necessary to create a unique Call Request packet.

For PAD calls that require PADPARM= values to override those on the root MCH, an MXT rather than an SVC should be used in place of a *dteaddr*i value. This is because the SVC REMOTE does not support the PADPARM= operand.

If PADPARM= values are not required, we recommend using an SVC rather than an MXT for call processing. The SVC control block uses less memory than the MXT. The SVC control block is X'178' (376) bytes in length while the MXT control block is X'480' (1152) bytes in length.

Note that when an MXT or SVC is specified in place of a *dteaddr*i value, the MXT identified by *mxt-name* will be ignored for the duration of the LU/VC session. However, if specified, this MXT will be used to supply VTAM parameters for the SLU during AMNF generation via the FASTRUN process. If this MXT is not specified, VTAM parameters come from the root MCH.

If the direction indicator at the end of the *dteaddr*i list is a 'T' instead of an 'O' (two-way support) and a *dteaddr*i list entry is a *rmtname*i instead of a *dteaddr*i, the named REMOTE will be used for inbound as well as outbound connections. If the direction indicator is an 'I' and a *dteaddr*i list entry is a *rmtname*i instead of a *dteaddr*i, the named REMOTE will be used for inbound connections only. For inbound calls, the DTEADDR= value is used like a *dteaddr*i value from *dteaddr*i list. If the *calling* DTE address from the inbound Call Request packet matches the DTEADDR= operand value, the SLU will be allocated for the inbound call.

The new syntax for the REMOTE SVC0= and SVC5= operands is as follows:

```
MCH1 REMOTE TYPE=MCH|XTP
            SVCx=(n,sluname/dteid1-dteid2-dteid3{T|O|I}[applid]
                                /mxtname/cud,...
```

The new syntax for the MRMT SVC0= and SVC5= parameters is as follows:

```
MRMT mchname SVCx=sluname/dteid1-dteid2-dteid3{T|O|I}[applid]/mxtname/cud
```

```
Where: dteidi = dteaddri (dd...dd)          (for T|O|I)
              = Xidnumi (Xdd...dd)          (for T|I)
              = <rmtnamei>                  (for T|O|I)
            x = 0 for SVC0 or 5 for SVC5
```

<u>CDF example to specify a *dteaddr*, X*idnum* and <*rmtname*> value for the same SLU:</u>

```
MCH1 REMOTE TYPE=MCH
            SVC0=(20,
                  MCH10011/1234-X1234-<SVC1>T01/MXT0,
                  :

SVC1 REMOTE TYPE=SVC
            DTEADDR=10981000,DCEADDR=20351000
            FAC=420909430707,CUD=C0000000
```

```
MXT0 REMOTE TYPE=MXT
            DTEADDR=10991000,DCEADDR=20361000
            FAC=420707430303,CUD=C0000000

MXT1 REMOTE TYPE=MXT
            DTEADDR=10991001,DCEADDR=20361001
            FAC=420808430303,CUD=C0000000
```

<u>MRMT example to modify the X*idnum* and <*rmtname*> value while leaving the *dteaddr* value alone:</u>

```
MRMT MCH1 SVC0=MCH10011/-X6789-<MTX1>
```

Note that 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 (1234 in this case).

<u>DRMT example to review MRMT change:</u>

```
DRMT MCH1 SVC0=MCH10011

RMTNAME  OPERAND  TYPE
MCH1     0084F4E8 MCH
         SVC0=0020
              MCH10011/1234-X6789-<MXT1>T01/MXT0
```

Notes: 1) Up to 3 *dteid*i values may be specified for any direction identifier (T|O|I). X*idnum*i values are only used for inbound connections while *dteaddr*i and <*rmtname*i> values can be be used for inbound and outbound connections. For inbound connections, the SLU is allocated if a match occurs for any *dteid*i value in the list.

2) The *rmtname*i in the *dteidi* list must be enclosed in non-equivalence symbols <>. The MXT|SVC addressed by a <*rmtname*i> operand is used for the duration of the call. Note that in this case, an MXT name coded after the *dteidi* list (/*mxtname*) is not used. It is only used when a *dteaddr*i value is specified as a *dteid*i list entry.

3) A *dd*...*dd* value prefixed with an ‘X’ is used as an IDNUM value to be matched against IDNUM value that is carried in the CUD field of an inbound Call Request packet. If a match occurs, the SLU is allocated for the inbound call. The X*dd*...*dd* value can contain up to 14 paired hex digits.

4) A *dd*...*dd* value without the ‘X’ represents a DTE address that is used as the *called* DTE address in an outbound Call Request packet OR is matched against the *calling* DTE address from an inbound Call Request packet. If a match occurs, the SLU is allocated for the inbound call. The *dd*...*dd* value can contain up to 15 decimal digits.

5) Multiple *dteid*i values may also be specified when the direction identifier is **I**. In this case, inbound *dteaddr*i or X*idnum*i matching will look at up to 3 values. If <*rmtname*i> is specified as a *dteid*i value, the *dreaddr*i will come from the DTEADDR= operand on the named MXT|SVC.

6) X*idnum* values are not supported in the DTEADDR= operand on a TYPE=MXT REMOTE. An X*idnum* value may only be specified as a *dteid*i value.

HNAS V2R3M0 New Features

V2R3M0 General Availability - 03-22-2004

Functional Enhancements:

- All appropriate V2R2M0 **APARs** have been incorporated as permanent fixes in HNAS V2R3M0 as of the general availability date.

- **SMP/E Product Installation Support Implemented for HNAS V2R3M0 edistributions**

  SMP/E product installation now available for HNAS 230 edistributions. The initial SMP/E product implementation was designed for installation using HNAS private global zone support although common shared global zone support was added to 230 effective 03-31-2004.

  The HNAS 230 product can now be installed using SMP/E or the default TSO Transmit format installation method. If you require SMP/E please indicate as such when you order the product.

- **Extended Diagnostic Event Reason Code support**.

  Extended diagnostic event reason codes are now displayed in various alert/alarm messages, trace entries and some console display messages. These extended codes will make it easier to define session connect and disconnect causes as well as improve general debugging.

- **CART= WTO support for Session Manager (Netview, TDSLink) Operator Consoles**.

  The CART= operand is now supported on the WTO macro to pass a token from MODIFY command back to the MODIFY sender in order to associate console commands with responses. This is used by Netview, TDSLink and other host products to solicit specific information from the HNAS console subsystem.

  The new ALRMCART= operand has been added to the BUILD definition statement so that alarm messages can be routed to a specific component like Netview or TDSLink. The ALRMCART= token is passed to the WTO service routine via the CART= operand for alarm messages only.

  Refer to the “**NETVIEW** routing support for asynchronous alarm messages as well as synchronous console command output...” later in this section for additional Netview support services information.

- **Improved Console and SYSPRINT output filtering.**

  Using new **PRNT *type*** and **SHOW CONS** start parameters/console commands, user can now control which specific types of output are directed at the SYSCONS or SYSPRINT interfaces.

- **SYSPRINT dataset recording improvements**.

  HNAS logic enhanced to better handle SYSPRINT *ddname* assignment and dataset full conditions.

The PRTLMT= operand on the BUILD definition statement limits the number of records that can be written to SYSPRINT while an HNAS ABEND EXIT eliminates SB37 SYSPRINT dataset full conditions. <See Problem Log Ref: 2002346A if additional information is required>.

Future enhancements will allow automatic switching of SYSPRINT datasets based on a list provided in the Configuration Data File. When a dataset becomes full, the next *ddname* in the SYSPRINT list will be used to continue the logging of events and or trace information.

- **SYSPRINT log filtering improvements**.

New HNAS start parameters and PRNT console command arguments have been added so that various informational messages (severity code of **I**) can be filtered from the SYSPRINT log based on the component type. HNAS starts with all component logging enabled. To turn off informational message logic for a particular component, the following start parameters may be used:

```
PRNTCNFG OFF - Inhibit logging of informational configuration messages.
PRNTLU OFF   - Inhibit logging of informational LU messages.
PRNTSYS OFF  - Inhibit logging of informational system messages.
PRNTTCP OFF  - Inhibit logging of informational TCP/IP messages.
PRNTVC OFF   - Inhibit logging of informational VC messages.
PRNTXOT OFF  - Inhibit logging of informational XOT messages.
PRNTXTP OFF  - Inhibit logging of informational XTP messages.
PRNTVTAM OFF - Inhibit logging of informational VTAM messages.
```

To turn on or off informational message logic for a particular component using the PRNT console command, you may enter the following:

```
PRNT CNFG ON|OFF - Toggle logging of informational configuration messages.
PRNT LU ON|OFF   - Toggle logging of informational LU messages.
PRNT SYS ON|OFF  - Toggle logging of informational system messages.
PRNT TCP ON|OFF  - Toggle logging of informational TCP/IP messages.
PRNT VC ON|OFF   - Toggle logging of informational VC messages.
PRNT XOT ON|OFF  - Toggle logging of informational XOT messages.
PRNT XTP ON|OFF  - Toggle logging of informational XTP messages.
PRNT VTAM ON|OFF - Toggle logging of informational VTAM messages.
```

- **Start parameter follower improvements**.

All HNAS start parameters may now accept ON and OFF as followers. ON is the default follower if none is specified after a parameter. For example, TRCPRNT and TRCPRNT ON are treated the same. OFF turns off the function that the parameter sets. For example, TRCVC OFF turns off the default TRCVC MINDATA setting.

- **XOT TAP= support improvements**.

XOT TAP= support now provides calling/called address, facilities and call user data parameters allowing **greater control** over **routing** and **filtering** of the special XOT call request packet used for tapping control. The tapping XOT calls can now be **selectively cleared** (providing the necessary tapping response) using the Cisco '**x25 route *dteaddr* clear**' configuration option. The tapping DTEADDR=, DCEADDR=, FAC= and CUD= fields are supported on the TYPE=XOT REMOTE for remotes with tapping enables (TAP=nn option).

- **TRCTRAP= diagnostic automation tools**

  TRCTRAP= support provides automation tools for improved debugging and trace suspension.

  Sometimes it is difficult to capture the cause of a problem in the HNAS internal trace table when tracing has to be stopped manually. In many cases, the only way to resolve a problem is to run HNAS traces. If a problem is intermittent, TRCPRNT 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 may be difficult and, in some cases, may even be impossible.

  HNAS has been modified to accept a new configuration operand and console command that will provide the ability to suspend internal tracing (and TRCPRNT) based on alarm message ID match, 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)...

  Note: Please refer to the Enhancements - Expanded Content heading at the end of this section for additional information regarding this new or enhanced feature.

  This new feature was introduced into 230 as Enhancement APAR 2300045/2300047.

- **NETVIEW** routing support for asynchronous alarm messages as well as synchronous console command output can be provided using the PFXWTO, PFXWTO CONS and SHOWCONS start parameters or the PFXWTO ON, PFXWTO CONS and SHOW CONS console commands in conjunction with the Netview ASSIGN command as follows:

  Use the PFXWTO start parameter or the PFXWTO ON console command to cause the NASNAME= operand value to prefix all alarm messages issued by HNAS.

  Use the PFXWTO CONS start parameter or the PFXWTO CONS console command to cause the NASNAME= operand value to prefix all console command output when SHOWCONS is in affect.

  Use the SHOWCONS start parameter or the SHOW CONS console command to cause all console command output to be routed to SYSCONS in addition to SYSPRINT.

  To allow Netview to filter on the NASNAME= operand value, you need to issue the following Netview ASSIGN commands:

  For alarm messages: `ASSIGN MSG=`*`nasname`*`,PRI=`*`opername`*

  For console output: `ASSIGN MSG=`*`nasname`*`,COPY=`*`opername`*

  Some improved features for this support was introduced into 230 as Enhancement APAR 2300006 and 2300026.

TRCTRAP= diagnostic enhancement

Some times a customer is unable to capture the cause of certain problems when systems operation is automated/unattended. Customers have reported that 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.

To avoid the need for an operator initiated SHUTDOWN or a forced ABEND, the TRCTRAP= operand logic has been modified to allow an action operand (,TRAPACTION=) as follows:

```
TRCTRAP=(ALRMLIST=(...),
         RCVLIST=(...),
         XMTLIST=(...),
         TRAPACTION={ALL|NONE|({SUSP|NOSUSP},{SNAP|NOSNAP})})
```

The action taken when a TRCTRAP 'hit' occurs currently defaults to TRAPACTION=SUSP which causes HNAS tracing and SYSPRINT trace logging to be suspended. The new TRAPACTION=SNAP operand will allow a snapshot of all HNAS control blocks and trace entries to be logged in SYSPRINT when a 'hit' occurs. If SUSP is required in addition to SNAP, both must be specified so that both functions are performed. For example:

TRCTRAP=(TRAPACTION=(SUSP,SNAP)) or simply TRCTRAP=(TRAPACTION=ALL).

If either SUSP or SNAP is specified by itself, only that action is performed on a hit. If the TRAPACTION= suboperand is omitted, SUSP is assumed which provides downward compatibility with operation under implementation of Enhancement APAR 2300045. If you wish to define trap filters but postpone any action on a trap hit until trace trapping is started or resumed using the TRCTRAP RSME, TRCTRAP RSMESNAP or TRCTRAP RSMEALL console commands, specify TRAPACTION=(NOSUSP,NOSNAP) or simply TRAPACTION=NONE.

> **Note**: The TRCTRAP console command has been modified to accept TRAPACTION=({SUSP|NOSUSP},{SNAP|NOSUSP}) as well as TRAPACTION={ALL|NONE} as an argument in order to provide equivalent functions for the TRCTRAP= configuration operand.

This new feature was introduced into 230 as Enhancement APAR 2300094.

- **WTO** routing codes can now be configured so that alarm messages can be routed to the **NETVIEW** NETLOG without also going to SYSLOG when SYSLOG is defined as a HARDCOPY console. 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.

Note: 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.

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.

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*).

Note: WTOROUTCDE(ALRM)= support was introduced into 230 under APAR 2300163.

Configuration Enhancements:

- Various **CDF Options** and **Parameters** were added or changed to allow the user greater flexibility when defining resources in V2R3M0:

- **Callout Session Connect Balancing Extension**

  Consecutive RTEOUT= entries with the same selection address can be used as a round robin group. Outbound calls will be balanced across the group. OPTIONS=BALANCERTEOUT is required.

- **Backup Router Support for LLC0/5 Callout**

  When an LLC0/5 callout fails (timeout or clear) HNAS will try to find an alternate path for the call by continuing to process the DTE address from the SVC0/5 entry in the RTEOUT= list. If another match in RTEOUT= is located the call will be tried on the associated router. After the entire RTEOUT= list has been processed the process is repeated for the second and third DTE address in the SVC0/5 entry (if present).

- **CUD=, FAC=, DCEADDR= and DTEADDR= (new) allowed on TYPE=XOT REMOTEs**

  These operands allow the fields of a Call Request packet to be provided by the user for the router TAP request. When these operands are omitted, the TAPping Call Request packet will contain no called and called DTE address, no facilities data and default call user data of 01000000HNASTAP.

- **OPTIONS=INHIBITBIDREJ** added for the TYPE=MCH|XTP REMOTE definition statement. This option sets **never reject BID** mode.

- **OPTIONS=NORTRBIDREJ** added for the TYPE=MCH|XTP REMOTE definition statement. This option will cause HNAS to **not generate 0814 for a BID reject.**

- **OPTIONS=ONEPIUINB** added for the TYPE=MCH|XTP REMOTE definition statement. This option will **force one PIU per bracket, inbound** (towards application) mode.

- **OPTIONS=REQSESSDELAY=*value*** added for the TYPE=MCH|XTP REMOTE definition statement. The ***value*** you specify is used to provide a delay between the UNBINDing of an LU and the REQSESS that solicits a subsequent BIND. The delay is needed by some applications to give them time to recycle their LU(s). In the past no delay was enforced which, in some cases, caused the REQSESS PIU to be rejected. You may code a ***value*** between 0 and 254. If omitted, 2 seconds is assumed.

- **OPTIONS=RESIDSTART=*decimal-start-value*** added for the TYPE=MCH|XTP REMOTE definition statement. This option allow users to **specify the decimal GATE resid (resource id) start *value***. X'FFFF' =>omitted (use default based on LCN0USED, PVC count).

- **INIT=({ACTIVE|IDLE},DELAYTIME=*minutes*,RETRYLMT=*count*)** added for the TYPE=XOT|XTP LOCAL definition statement.

  Where: *minutes* can range from 0 to 62 and *count* can also range from 0 to 62. The INIT= operand values can also be modified using the MLCL.

  The DELAYTIME value is used to provide a delay after a TCPIP BIND failure before the BIND is retried. If 0 is specified, no delay is provided. The RETRYLMT value is used to count the number of BIND failures before the LOCAL is taken offline (INIT=IDLE forced). If 0 is specified, the BIND is retried indefinitely. If the INIT= operand is omitted from a LOCAL, INIT=(ACTIVE,DELAY-TIME=5,RETRYLMT=0) is assumed.

  Note that when the BIND ends successfully, you will continue to get the following standard alert:

  NAS2202I SERVER=*ipaddr*(*port*) SOCKID=*xxxx* PCEID=*xxxx* NAME=*rmtname*
  NAS2020I SERVER INITIALIZATION COMPLETE

  If all BINDs fail and a RETRYLMT value is specified, you will get the following new alert:

  NAS2202W SERVER=*ipaddr*(*port*) SOCKID=*xxxx* PCEID=*xxxx* NAME=*rmtname*
  NAS2020W SERVER INITIALIZATION FAILED, LOCAL VARIED OFFLINE

  In all cases the NAS0001I HOST NAS INITIALIZATION COMPELE, ALL FUNCTIONS00 READY message will be withheld until all LOCALs come active or a failing LOCAL is taken offline. If INIT=IDLE is specified, the LOCAL is offline when HNAS is started and no BIND is attempted.

  Additional information on this support is provided in Chapter 5, section Migration - V2R3M0, under Item (29).

- **IDLETO=*minutes*** added for non TYPE=XOT REMOTE definition statement. This support was previously provided as a global value on the BUILD definition statement.

- The HNAS CDF **configuration error summary** has been modified to display the count of default messages issued. In prior releases, both default and warning message counts were combined because the HNAS return code for their respective message severity was the same.

- The HNAS **configuration error processing** has been modified to display a common message ID (**NAS1999W|E|S**) marker whenever a configuration error message is generated that sets RC= 4|8|12. This will allow error messages to be quickly located in the HNAS SYSPRINT log using find or search tools (i.e. find 'NAS1999'). In addition, HNAS will now withhold generating configuration default messages (NAS1xxxD) if any configuration error message has been previously generated. Since HNAS processing will terminate after the CDF scan is completed due to RC>4, no default action can be taken, hence, no reason to generate default messages. The HNAS CDF scan no longer generates a RC-4 return code for (D)efault messages which further reduces completion code viewing requirements.

- The **FASTRUN** process will now propagate VTAM operands that are specified on a TYPE=MXT REMOTE definition statement if that MXT is associated with a SLU in the LUNAME= operand list on a TYPE=SPU REMOTE definition statement. If no MXT is associated with an SLU entry, the VTAM operands are taken from the root TYPE=SPU REMOTE definition statement as in previous releases. In this way, the generated APPL statements for each SLU can have different VTAM operands (for example, MODETAB, DLOGMOD, etc.).

- The **FASTRUN** process will now allow the name for the AMNF VBUILD statement to be specified using the APPLNAME= operand rather than NASNAME= operand on BUILD definition statement.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300007.

- The **configuration** process will now accept a pound sign (#) in record column one (1) to identify a comment line that will be logged in the HNAS SYSPRINT file. This will allow you to place block comments in front of definition statements or operands that will be listed in SYSPRINT when the CDF is scanned. Contrast this to standard CDF comment record that starts with an asterisk (*) in record column one which will not be listed in SYSPRINT.

- The new **PFXWTO CONS** start parameter causes HNAS console output to be prefixed using the existing **NASNAME=** value. This enhancement provides a unique message id that can be used for customer automation processes. This processing requires that the **SHOW CONS ON** option must also be in affect. This processing does not affect alarm output.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300006.

- The **PFXWTO *text*** start parameter may be specified to cause the *text* value to be appended to the beginning of each alarm WTO in lieu of the NASNAME= operand value. This same text is used to prefix console output if the PFXWTO CONS option is also in affect. Up to 8 non-blank characters may be specified.

Specify **PFXWTO *text* PFXWTO OFF PFXWTO CONS** if you only want console output to be prefixed with the *text* value.

Note: This new feature was introduced into 230 as Enhancement APAR 2300026.

In 220 and earlier versions, **PRNT** is a default state (not a start parameter) that enables global SYSPRINT logging. The state can be altered using the **PRNT ON** or **PRNT OFF** console command.

- **OPTIONS=PFXDCEADDR** added for the TYPE=MCH|XTP REMOTE definition statement. This option specifies that the value coded for the **DCEADDR=** operand is to prefix (appended in front of) the ***calling*** DTE address in a GATE **outbound** Call Request packet when a ***calling*** DTE address (or subaddress) is provided by the CTCP. If no ***calling*** DTE address is supplied by the CTCP (field is null), PFXDCEADDR is treated as REPDCEADDR. TYPE=MCH and GATE=GENERAL must also be specified.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300052.

- **OPTIONS=LLC0CTCPCHK** and **OPTIONS=LLC5CTCPCHK** added for the TYPE=MCH REMOTE definition statement. This new option allows control of Call Accept logic for LLC0 and LLC5 sessions using a CTCP (as is done with DATE). When the LLC0 or LLC5 check option is set, inbound LLC0/5 calls will be cleared with a diagnostic of 254 if the first CTCP in the LUNAME= list is not active (the call request is not sent to the CTCP). These options when used in conjunction with a Transpac option will retry the call request using different addresses. This allows sessions to be switched to different MCHs.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300060.

- **CONLMT=0** is now allowed on the BUILD definition statement. This new specification allows remote console access to be barred and also suppresses default console configuration messages that are only applicable when remote console access is allowed. Prior to APAR 2300061, if CONPSWD= is omitted, the following configuration warning message is generated:

  ```
  NAS1101W BUILD CONPSWD OMITTED, REMOTE CONSOLE ACCESS WILL
           NOT BE ALLOWED
  ```

  With APAR 2300061 applied, when CONPSWD= omitted but CONLMT=0 coded, this warning message is suppressed. This support was added so that a customer, who may not want remote console access, no longer has to omit the CONPSWD= operand. This new option also provides users with a means of disabling remote console access without having to remove the sample CONSOLE MCH definition in the CDF template that is typically provided by Comm-Pro or their HNAS Business partners in sample CDF definitions. While start parameter RMTCONS OFF will also disable remote console access, some customers will want to disable it in the CDF via CONSLMT=0.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300061.

- Bidirectional **Twoway** (Callin/Callout) VC support can now be enabled for PCNE and PAD resources via **SVC0=** or **SVC5=** TYPE=MCH REMOTE definition statements.

The SVC0= and SVC5= operand logic has been modified to accept a **T** value (for **T**woway) following the DTE address or IDNUM value so that the same SLU can be used for callin and callout connections. When a **T** is detected, the ACB for the SLU is opened so that it can be acquired (bound). If an inbound call arrives before the SLU is bound and the calling DTE address matches one of the called DTE addresses associated with the SLU or the CUD field contains an IDNUM value that matches one of X*idnum* values associated with the SLU (there can be up to 3 DTE addresses and/or X*idnum* values specified for a twoway SLU), the SLU will be allocated to the inbound VC.

If a BIND arrives prior to an inbound call, the SLU will be processed in the same fashion as an SLU defined with the **O** (for callout) option. In other words, if the SLU is bound (acquired) before any inbound Call Request is received, the DTE addresses in the list will be used (successively) for outbound call attempts as the called DTE address. Any X*idnum* values in the list will be skipped.

Note: This logic change was introduced into 230 under APAR 2300083.

- **PVC=** ***mxtname*** support added to the TYPE=MCH REMOTE definition statement so that user defined packet and window size values can be provided in the HNAS PVC Setup packet when PVC= ***rmtname*** option is enabled.

  Note: This new feature was introduced into 230 as APAR 2300107.

- **SVC*i*=(*pfxlu*,*sfxst*,*vclmt*)** coding support added to the TYPE=MCH|XTP REMOTE definition statement so that user can control the generation of SLU names based on a prefix value (*pfxlu*), suffix start value (*sfxst*) and SLU count value (*vclmt*).

  Currently, the first SVCi= entry is the *vclmt* and the subsequent entries are SLU names. Missing SLU names are replaced by default SLU names which are created using the first 4-characters of the REMOTE name followed by a 0|4|5 for the LLC type followed by the SVCi= index number (001, 002, 003, etc.).

  Primary: SVC*i*=(*vclmt*,*slunm*1,*slunm*2,...)
Alternate: SVC*i*=(*pfxlu*,*sfxst*,*vclmt*)

  The primary and alternate specifications will both be accepted. The alternate specification is appropriate when no additional suboperands need to be associated with each SLU name. For the alternate specification:

  The *pfxlu* value must be the first SVC*i*= suboperand and may be any valid assembler language symbol up to 7-characters in length starting with either an alpha character (A,B,C,...,Z) or an accepted special character (@, #, $ or %). This suboperand is REQUIRED to indicate that the alternate specification is being used.

  The *sfxst* value must be the second SVC*i*= suboperand and must be a hexadecimal number (without the framing characters X'') between 0 and F when the *pfxlu* length is 7, between 0 and FF when *pfxlu* length is 6, ..., between 0 and FFFFFFF when the *pfxlu* length is 1. If *sfxst* is omitted (,,), a default value of 0 will be used.

  The *vclmt* value must be the third SVC*i*= suboperand and must be a decimal number between 0 and 511 (the SVC*i*= array size). This suboperand is REQUIRED. The SLU names that HNAS generates from the *pfxlu*, *sfxst* and *vclmt* values will always be 8-characters in length with zero (0) pad characters added as required between the last *pfxlu* character and the suffix value.

Examples:

If SVC0=(TT,1,3) is specified, the generated SLU names would be TT000001, TT000002, TT000003.

If SVC0=(TTTTT,0,3) is specified, the generated SLU names would be TTTTT000, TTTTT001, TTTTT002.

Note: This new feature was introduced into 230 as APAR 2300138.

- **TRAN=NPSIEVEN|NPSIODD|NPSIMARK|NPSISPACE** values are now accepted in order to allow standard NPSI translate tables to be specified in addition to standard HNAS translate tables. Prior to APAR 2300152, NPSI translate tables could only be included in HNAS by renaming the MCHTTBL1 member in the HNASOBJX library to MCHTTBLS then re-linking HNAS. This allowed the standard NPSI translate tables to be used in lieu of the standard HNAS translate tables using the existing TRAN= operand values of EVEN|ODD|MARK|SPACE. The new TRAN= values now permit NPSI as well as HNAS translate tables to be coresident and CDF configurable. NPSI tables are more appropriate and are even required for some host applications like TPE.

  Note: This new feature was introduced into 230 as APAR 2300152.

- Allow **RTEIN=** processing to be controlled by source calling address (**=>S**) in addition to the existing target called DTE address (**=>T**) in a manner similar to that used for the RTEOUT= operand. Please refer to remote type LOCAL RTEIN= parameter for additional information.

  Note: This new feature was introduced into 230 as APAR 2300176.

- **OPTIONS=BALANCERTEIN** enhancement to RTEIN= processing allows inbound calls to be distributed across multiple MCHs using a round robin session connect balancing scheme. Please refer to remote type LOCAL OPTIONS=BALANCERTEIN and RTEIN= operand for additional information.

  Note: This new feature was introduced into 230 as APAR 2300176.

Alert/Alarm Message and Clear/Reset Code Enhancements:

- **Message filtering** enhancement.

  Informational messages can now be filtered from the HNAS SYSPRINT log file (SYSOUT) in addition to the system console log. In conjunction with the SHOWERR start parameter (or SHOW ERR console command), you may now specify the following options to prevent certain informational messages from being written to the HNAS SYSPRINT log file.

Message ID	Component	Start Parameter	Console Command
NAS0*xxx*I	System (General)	PRNTSYS OFF	PRNT SYS OFF
NAS1*xxx*I	Configuration	PRTNCNFG OFF	PRNT CNFG OFF
NAS2*xxx*I	TCP/IP	PRTNTCP OFF	PRNT TCP OFF

NAS3*xxx*I	VTAM	PRTNVTAM OFF	PRNT VTAM OFF
NAS4*xxx*I	LU	PRTNLU OFF	PRNT LU OFF
NAS5*xxx*I	VC (Common)	PRTNVC OFF	PRNT VC OFF
NAS6*xxx*I	XTP	PRTNXTP OFF	PRNT XTP OFF
NAS7*xxx*I	XOT	PRTNXOT OFF	PRNT XOT OFF
NAS8*xxx*I	VC (QLLC)	PRTNVC OFF	PRNT VC OFF
NAS9*xxx*I	System (Auth)	PRTNSYS OFF	PRNT SYS OFF

- **Julian Date (*nnn*)** added to **Alert** Message prefix for improved identification of the day the event occurred.

- **NAS2021W Alert** Message was added to improve notification regarding server notification. The message will be generated when HNAS cannot connect to the stack after the retries that are specified in the REMOTE INIT operand have been reached. See section Configuration Enhancements, **INIT=({ACTIVE|IDLE}** description in this document for additional information.

- **NAS2601E** and **NAS2602I Alert** Messages were added to improve XOT socket availability status notification. These messages extend HNAS awareness of socket availability beyond the stack to the router. The socket slowdown messages are as follows:

  NAS2601E SOCKET POOL DEPLETED, SOCCNT=ddddd SOCLMT=ddddd ',

  This message is issued when the number of sockets in use (SOCCNT) reaches 80% of the maximum number of sockets for a server (SOCLMT)

  NAS2602I SOCKET POOL RESTORED, SOCCNT=ddddd SOCLMT=ddddd ',

  This message is issued when the number of sockets in use (SOCCNT) returns to 60% of the maximum number of sockets for a server (SOCLMT)

- **NAS8000I Alert** Message was modified to improve QLLC call setup status notification.

  NAS8000I CLIENT=*iii.iii.iii.iii*(*port*) SOCKID=*sockid*
  PCEID=*pceid* NAME=*rmtname*
  NAS8000I VC *vcb* STARTING SESSION ON MCH *mchname* (*count*)
  NAS8000I *direction* CLGADDR=*lddd...ddd* CLDADDR=*lddd...ddd*

  This message is issued when a QLLC session starts for the VC associated with the MCH named ***mchname***. The VCB address ***vcb*** and active VC connect counter ***count*** are also displayed. The ***count*** value identifies the number of virtual circuits connected to the denoted MCH. The ***direction*** value identifies the call as being INBOUND to HNAS or OUTBOUND from HNAS. The Calling DTE address (CLGADDR) and Called DTE address (CDLADDR) are also displayed. For both, ***lddd...ddd*** is the decimal DTE address (***ddd...ddd***) prefixed with the hexadecimal single digit address length (***l***).

  Note: This enhancement was introduced into 230 as Enhancement APAR 2300039.

- **NAS7718T, NAS7719T and NAS7798T Special Trace Alert** Message added to provide formatted content of inbound and outbound call request packets. The special **NAS*innn*T** trace entry alert message types are only written to sysprint not SYSCONS (system operator console).

  Note: This new feature was introduced into 230 as Enhancement APAR 2300046.

- **NAS3703W** (NOTIFY) and **NAS3704W** (CLEANUP) VTAM Alert Messages were added to improve diagnostic capabilities of REQSESS related error conditions by providing sense code information. Please refer to the VTAM alert messages section for a description and layout of these messages.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300054.

- **NAS7715W** (CALL REQ TO MCH FAILED) alert message now contains the extended diagnostic event reason codes (DIAGX=xxxx) values to improve diagnostic capabilities. This message wasn't included in the group of alert messages that were provided with this support when HNAS 230 was initially released.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300053.

- **NAS7713W** SECOND CLEAR FROM ip-addr(port) ON rmt-name MCH mch-name LU lu-name alert message added improving event reporting and problem determination.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300085.

- **Clear** Request **diagnostic codes** were added to improve event reporting and fault isolation:

  Clear diagnostic code **221 x'DD**' - Timeout. CTCP did not UNBIND after receiving clear
  Clear diagnostic code **222 x'DE**' - Second clear received from router for VC

  Note: These new diagnostic codes introduced into 230 under APAR 2300085.

- **NAS7797W** CALL FROM ip-addr. PCE HAS VC AT vc-addr LCST=st LU=lu-name LU ADDR=lu-addr alert message added indicating that HNAS may have failed to properly terminate an XOT TCP/IP session with a router. Please contact customer support if this alarm is received.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300092.

- **NAS3797I** LU sluname RECEIVED BIND FROM PLU pluname PVC messages added so that users could easily identify when an LU with a PVC session receives a BIND from the PLU. Because PVCs are always active, HNAS tries to establish a session with the PLU using a REQSESS macro (asks PLU for a BIND) issued by timer logic. There can be many hours between the NAS3798I session starting message and the time when the PLU sends a BIND to start the session.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300118.

- **NAS4706W** LU sluname REJECTING BIND FROM PLU pluname SENSE=xxxx message was added for all LLC-n types so that users could easily identify the cause of a BIND failure. This message is issued when HNAS rejects a bind command sent by the PLU. The sense code indicating the cause of the reject.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300118.

- **Clear** Request **diagnostic code 211 x'D3**' was added to mirror NPSI's handling of a SIGNAL PIU sequence. When HNAS receives a SIGNAL PIU a +RSP will be returned to the PLU. For LLC5 sessions an IOB (indication of break) Q packet will be sent to the remote. For other LLC types SIGNAL does not cause X25 side activity. This processing conforms to NPSI operation.

  Note: This new diagnostic code was introduced into 230 under APAR 2300140.

- **NAS3705W** LU *lu-name* REJECTING message added indicating that PIU is being rejected. Up until now there was no alert message issued to denote when the PIU is rejected so it is difficult to determine the reason for the session end.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300144.

- **NAS7774W** PVCSETUP FAILED **-** REMOTE rmt-nm CLOSED TCP SESSION... message added indicating that the remote responded to a HNAS PVC setup packet with a socket close (FIN).

  Note: This new feature was introduced into 230 as Enhancement APAR 2300151.

- **NAS7718T, NAS7719T and NAS7798T Special Trace Alert** Message added to provide formatted content of inbound and outbound PVC Setup packets. The special **NAS*innn*T** trace entry alert message types are only written to sysprint not SYSCONS (system operator console).

  Note: This new feature was introduced into 230 as Enhancement APAR 2300151.

- HNAS **PVC Setup Status Codes** were reassigned improving router setup retry processing. Additional information is available in Chapter 5 - Migration, under the 230 section concerning this topic.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300151.

- **NAS7795T Special Trace Alert** Message added providing an alert message trace record for all inbound and outbound clear request packets. Logging for these new message types is enabled using the new TRCMCH ICLR|OCLR option. The special **NAS*innn*T** trace entry alert message types are only written to sysprint not SYSCONS (system operator console).

  Note: This new feature was introduced into 230 as Enhancement APAR 23000153.

- **NAS251xM Monitor Alert** Messages added providing information on TAP Keep Alive activity for REMOTE TYPE=XTP|XOT) resource. Console command **MON TAP {ON|OFF|ALLON|ALLOFF}** or Start Parameter **PARM=MONTAP** enable generation of these message types. TAP status log entries are written to SYSPRINT only.

Note: This new feature was introduced into 230 as Enhancement APAR 2300155.

- **NAS4710W** lu-nm LU lu-addr SENDING DIAG PKT *text* BFR NEXT trace record now provided as an alert message. Message is generated when HNAS sends a diagnostic packet to the CTCP. BFR NEXT indicates the buffer that triggered the error is displayed in SYSPRINT while the error indication **text** values are shown below:

  INVALID CTCP GATE CMD BYTE - HNAS has received an invalid GATE message from a GATE CTCP control session PLU.
  FMD INV ON FC GATE CTL SES - HNAS has received an FMD PIU from a GATE FC PLU. FC GATE control session do not send FMD PIUs.
  INVALID CTCP DATA SES CMD BYTE - HNAS has received an invalid GATE message from a GATE CTCP data session PLU.
  GT FC LN ER - HNAS has received a PIU from a GATE FC PLU that is too short to be valid.
  GT DS LN ER - HNAS has received a PIU from a GATE PLU that is too short to be valid.

  The most likely cause of NAS4710W is that the PLU is not a GATE CTCP.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300160.

- **NAS3705I** LU lu-name REJECTING cmd #seq SENSE=bbbbbbbb BID alert message is now issued when HNAS rejects a PIU from the PLU SENSE=0813xxxx or 0814xxxx. These values indicate that the PLU and the HNAS SLU are attempting to start a bracket at the same time. These rejects can be considered normal so the severity code is ‘I’ (follows ‘NAS3705’).

  OPTIONS=(INHIBITBIDREJ,NORTRBIDREJ) may be coded to reduce the number of NAS3705I alerts with SENSE=0813/0814.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300177.

- **NAS4707W** sluname GENERATING ERR/INFO PACKET FOR CTCP alert message is issued when HNAS receives an invalid packet on a GATE control or data session LU. In addition to the alert message an ERROR/INFORMATION REPORT message (not a DIAGNOSTIC message) will be sent to the CTCP on the control session LU.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300178.

- **NAS4708W** GATE FC CTL SES LU luname CLEARED BY CTCP alert message is issued when HNAS receives a CLEAR packet on an FC control session LU a NAS4708W alert message will be issued and a CLEAR-CONFIRM will be returned to the CTCP. Data session LUs will not be affected.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300178.

- **NAS4709W** REMOTE rmt-nm lu-nm LU lu-addr LUIQ TIMEOUT, LUIQ BFR CT=xxxx message is issued when HNAS has a message from the remote that has been waiting for delivery to the PLU for a period of 4 minutes. The alert indicates that there is something wrong with the VTAM session between the HNAS SLU and the PLU. After the ALERT the LU's VTAM session is ended (PLU will receive a NOTIFY). If there is an associated VC session it is cleared with DIAG=212.

  Clear Request **diagnostic code 212 x'D4**' was added to report this condition.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300180.

- **NAS3797I** alert message is now generated when a GATE control session is bound. Previously this alert was only used for BINDs received for PVC LUs. This will eliminate the requirement to view the HNAS DLU command display output to confirm that the GATE control session is active.

  `NAS3797I LU` ***`lu-nm`*** **`RECEIVED BIND FROM PLU`** ***`plu-nm`***

  Note: This logic change was introduced into 230 under APAR 2300201.

Console and Trace Enhancements:

- **Error reporting enhancement** provides more targeted information when a command is rejected due to a syntax, specification or consistency error.

- A **resource name** may now be specified on the right side of a command without a command modifier like RNM=, LNM= or LUNM=. The console command parser will associate the specified name with the appropriate command modifier based on the type of resource specified. This enhancement is provided to further simplify console command syntax.

- The **DMAP APAR** command has been modified so that it automatically executes at initialization time *with no delays*. The output of the command is logged in the HNAS SYSPRINT so that maintenance 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 new **DNAS APAR** command. Note that you can still use the DMAP APAR command to display APARs but command output is in module name order rather than APAR ID order.

  Note: This new feature was introduced into 230 as APAR 2300004 and 220 as APAR 2200079.

- The **DNAS command** has been modified to display the APAR=*apar-id* of the latest maintenance applied by the customer in addition to the *apar-id* that was incorporated when the HNAS distribution was created. The AUTH=*month-count* value was also added to the DNAS display output to provide the number of months established for the trial period.

  Additionally, a new argument (APAR) has been to the DNAS command so that a table, created at initialization time by the **DMAP APAR** command, can be displayed. The **DNAS APAR** command will display all APARs on the HNAS system in APAR ID order while the DMAP APAR command displays all maintenance in module name order.

Note: This new feature was introduced into 230 as APAR 2300004 and 220 as APAR 2200079.

- The **TRCLU command** will now start or stop tracing for all SLUs associated with a TYPE=SPU REMOTE definition statement that is identified by the RNM= command modifier. This is an extension to the logic that exists in V2R2M0 for TYPE=XTP|MCH REMOTE definition statements.

- The **DLU command** will now display all SLUs associated with a TYPE=SPU REMOTE definition statement that is identified by the RNM= command modifier. This is an extension to the logic that exists in V2R2M0 for TYPE=XTP|MCH REMOTE definition statements.

- The **DPCE command** will now display the PCE for a TYPE=SPU REMOTE definition statement that is identified by the RNM= command modifier. This is an extension to the logic that exists in V2R2M0 for TYPE=XTP|XOT|MCH REMOTE definition statements.

- The **DPCE, TRCDISP, TRCIO, TRCBFR, TRCDATA and STATS commands** have been modified to accept **CONS|TCP|TMR|UTIL|XOT|XTP** as a command argument so that only PCE resources of the specified type will be operated on.

- The **DLU, DVC and DPCE commands** have been modified to accept **TRACE=YES|NO** as a command argument so that only resources that *are or are not* being traced will be displayed. This is useful for finding all resources that are actively being traced.

- The HNAS **PING command** has been added to the console subsystem that will allow the console operator to check on the existence of a router on the IP network as well as a specific MCH on the router. The HNAS PING command is not a traditional IP PING command but rather an **X.25 PING**. The HNAS PING command requires an IP address to identify a router and an X.25 DTE address to identify the MCH link. These two pieces of information allow HNAS to establish an IP connection to the router and then an X.25 (XOT) connection via the MCH link. An indication of success or failure for the PING request is reported back to the console operator.

  Note: Please refer to the Enhancements - Expanded Content heading at the end of this section for additional information regarding this new or enhanced feature.

- The **MRMT command** was enhanced to allow users to modify QLLC (LLC3) SLU definitions in the LUNAME= operand of TYPE=SPU REMOTE definition statements via privileged local (SYSCONS) or remote console access.

  Note: Please refer to the Enhancements - Expanded Content heading at the end of this section for additional information regarding this new or enhanced feature.

- The **MRMT command** has been modified to permit updating of the INIT= operand of a TYPE=XOT|XTP REMOTE definition statement. Enter:

  RNM=*rmtname* MRMT INIT={ACTIVE|IDLE}

If the ACTIVE|IDLE value is changed, a V RMT ON|OFF command is propagated.

- The **MLCL command** has been modified to permit updating of the INIT= operand of a TYPE=XOT|XTP LOCAL definition statement. Enter:

  LNM=*lclname* MLCL INIT={ACTIVE|IDLE},DELAYTIME=*minutes*,RETRYLMT=*count*

  Where: *minutes* can range from 0 to 62 and *count* can also range from 0 to 62.

  Note: For the console command, the INIT= argument does not use opening and closing parens.

  If the ACTIVE|IDLE value is changed, a V LCL ON|OFF command is propagated.

- The **VARY command** has been modified to allow a LOCAL to be varied offline or online just like a REMOTE. The VARY command now has the following format:

  LNM=*lclname* V LCL {ON|OFF} **or** RNM=*rmtname* [RMT] {ON|OFF}

  If you vary a LOCAL (server) offline when it was previously online, subsequent calls to its HOME IP address will be cleared by the router because the server will not be LISTENing on its IP address. If you vary the LOCAL back online, subsequent calls will then be accepted. This is a nice feature for quiescing REMOTE access to HNAS without stopping it. Active REMOTE connections will remain active until terminated by either end.

- The **DTRC command** has been modified to display the WRAPCNT=*wrap-cnt* counter in the trace table header. The wrap count identifies the number of time the trace table has wrapped.

- The **DLU**, **DMCH** and **DPCE commands** now provide support for the **SHOWNAME|SHOWADDR** parmlist options while the **DVC command** now provides support for the **SHOWNAME|SHOW-ADDR| SHOWIPAD|SHOWCGAD|SHOWCDAD|SHOWVCST|SHOWLUST** parmlist options. These new **SHOW*type*** arguments provide for improved console command usability and resource status.

- The **TRCLU command** now produces trace entries providing keywords (BIND, SDT, UNBIND, etc.) improving search capabilities.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300010.

- The **DMCH command** now accepts **FMT3** as an argument which will cause 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. This command is currently provided under Local console operation. Remote console support will be included in HNAS 231.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300016.

- The **PFXWTO command** now accepts ***name*** as an argument which will allow alarm messages and console output to be prefixed with a user defined name other than the current NASNAME= option. This new feature can also be enabled as a parmlist option (PARM='PFXWTO *name*').

  Note: This new feature was introduced into 230 as Enhancement APAR 2300026.

- The **TRCMCH ICR and TRCMCH OCR commands** (and **CONCMDQ=**) trace logic was enhanced to provide improved formatting of inbound and outbound call request packet content. The information is provided in special NAS7718T, NAS7719T and NAS7798T alert trace messages that are written to sysprint only.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300046.

- The **MLCL command** has been modified to permit an insert/delete index to be specified for the RTEIN= and RTEOUT= operands.

  Prior to 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)
```

Note: The RTEIN/RTEOUT insert/delete index logic was introduced in 230 as Enhancement APAR 2300056.

- The **DNAS command** has been modified to display a list of missing *apar-ids* if the 'latest' *apar-id* is different than the *apar-id* that was incorporated when the HNAS distribution was created. For example, if you were shipped HNAS with APAR 2300059 incorporated and you then installed APAR 2300062 separately, the following would be displayed if you did not also install the 'pre-reqs' for APAR 2300062:

```
HNAS --> VERSION=V2R3M0 HOST=OS390|ZOS ASMDATE=08/02/04
         CREATED AT 14:11:31 ON 07/23/2004
         CREATED WITH MAINTENANCE THROUGH APAR 2300059
         MOST RECENT MAINTENANCE APPLIED IS APAR 2300062
         SHIPID=0000000000099999 AUTH=00
         APARID  MODULE   (MISSING MAINTENANCE)
         2300060 MISSING
         2300061 MISSING
```

If there is no missing maintenance, that is, if there are no holes in the *apar-id* values from the shipped *apar-id* value to the latest *apar-id* value (all pre-reqs were also installed), the following is displayed:

```
HNAS --> VERSION=V2R3M0 HOST=OS390|ZOS ASMDATE=08/02/04
         CREATED AT 14:11:31 ON 07/23/2004
         CREATED WITH MAINTENANCE THROUGH APAR 2300059
         MOST RECENT MAINTENANCE APPLIED IS APAR 2300062
         SHIPID=0000000000099999 AUTH=00
         APARID  MODULE   (MISSING MAINTENANCE)
         NOTHING MISSING
```

Note: The missing *apar-id* logic is only executed when DNAS is entered with no arguments.

Additionally, the DNAS APAR command now accepts an *apar-id* value to restrict the displayed applied maintenance to the specified *apar-id only.* For example: **DMAP APAR 2300062** will produce the following display:.

```
HNAS --> VERSION=V2R3M0 HOST=OS390|ZOS ASMDATE=08/02/04
         CREATED AT 14:11:31 ON 07/23/2004
         CREATED WITH MAINTENANCE THROUGH APAR 2300059
         MOST RECENT MAINTENANCE APPLIED IS APAR 2300062
         SHIPID=0000000000099999 AUTH=00
         APARID  MODULE   (APPLIED MAINTENANCE)
         2300062 CONSDNAS
```

```
CONSHELP
NASCONS
```

Note: This new feature was introduced into 230 as APAR 2300062.

- The **DNAS command** now accepts **USERMODS** as an argument which will cause the command to list any custom-user-mods applied to the system. The list includes the Problem Number or unique Customer Fix ID as well as the modules affected by the custom enhancement. Prior to this new display users had to search through the DNAS APAR or DMAP APAR display output for usermods intermingled in with standard APARs. Please refer to the console command description for additional information.

  Note: This new feature was introduced into 230 as APAR 2300065.

- The **DNAS APAR** and **DMAP APAR commands** now optionally display APARID's HNASMACX and/or HNASOBJX 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 information.

  Note that the majority of customers don't have custom enhancements (CustomUserMods) although this enhancement APAR was developed to improve their support services.

  Note: This new feature was introduced into 230 as APAR 2300069.

- The **MRMT command** processor has been modified to allow PCNE/GATE/PAD SLU names to be added, deleted or changed for the SVC0=/SVC4=/SVC5= operands on TYPE=MCH|XTP REMOTE's in a manner similar to what is now done for the LUNAME= operand on a TYPE=SPU REMOTE.

  Note: This logic change was introduced into 230 under APAR 2300084.

- The **SHOW MORE command** (and **SHOWMORE start parameter**) were added providing expanded TCP/IP stack event debugging.

  Note: This logic change was introduced into 230 under APAR 2300088.

- The **TRCALL command** was modified to remove the PCE trace function for the ON|OFF arguments. The processing for the ON and OFF arguments is now treated identically to the STRT and STOP arguments, respectively. The PCE trace function is now provided by the new **TRCPCE** command.

  Note: This logic change was introduced into 230 under APAR 2300110.

- The **DNAS command** has been modified to display the distribution type (SMP/E or NON-SMP) on the header line. In addition, up to 2 new lines can be displayed to show the name of an optional custom MACLIB and/or OBJECT library that was used to create a non-SMP distribution.

```
HNAS --> VERSION=V2R3M0 HOST=OS390|ZOS ASMDATE=02/24/05 DIST=NON-SMP
         CUSTMAC=COMM1.TEST.HNASMAC
         CUSTOBJ=COMM1.TEST.HNASOBJ
         CREATED AT 14:11:31 ON 02/26/2004
         CREATED WITH MAINTENANCE THROUGH APAR 2300000
         MOST RECENT MAINTENANCE APPLIED IS APAR 2300113
         SHIPID=9999999999999999 AUTH=00
         APARID   MODULE  (APPLIED MAINTENANCE)
         2300113  CONSDPRM
                  CONSLNCT
```

Note: This logic change was introduced into 230 under APAR 2300113.

- The **VARY command** has been modified to allow specific sockets or ranges of sockets to be varied offline or online. Please refer to the Console Subsystem guide for additional information and command syntax.

Note: This logic change was introduced into 230 under APAR 2300123.

- The **TRCMCH ICR** and **TRCMCH OCR command** trace logic was enhanced to provide improved formatting of inbound and outbound PVC setup packet content. The information is provided in special NAS7718T, NAS7719T and NAS7798T alert trace messages that are written to sysprint only.

Note: This new feature was introduced into 230 as Enhancement APAR 2300151.

- The **DVC command** has been modified to show whether or not a PVC setup exchange has occurred (display field VCOPT **D**=setup not complete, **I**=pvc received reset link/device inoperative and **N**=setup completed. Additional information is available in the DVC section under VCOPT in the HNAS Console subsystem guide.

Note: This new feature was introduced into 230 as Enhancement APAR 2300151.

- The **MRMT command** has been modified to permit updating of the TRAN= operand of a TYPE=MCH|XTP REMOTE definition statement. Enter:

```
RNM=rmtname MRMT TRAN={NO|USER|EVEN|ODD|MARK|SPACE|
                       NPSIEVEN|NPSIODD|NPSIMARK|NPSISPACE}
```

Note: This new feature was introduced into 230 as Enhancement APAR 2300152.

- The **TRCMCH ICLR** and **TRCMCH OCLR commands** were implemented to produce trace messages for all inbound and outbound clear request packet activity. The information is provided in the NAS7795T alert message (written to SYSPRINT only).

Note: This new feature was introduced into 230 as Enhancement APAR 2300153.

- The **MON[ITOR] command** has been modified to accept **TAP {ON|OFF|ALLON|ALLOFF}** as arguments for the purpose of start and stopping *local* (individual REMOTE TYPE=XTP|XOT) or *global* (all REMOTE TYPE=XTP|XOT) TAP (Keep Alive) monitoring. TAP status log entries are written to SYSPRINT only and are identified by alert message IDs of the form NAS251xM.

  Note: This new feature was introduced into 230 as Enhancement APAR 2300155.

- The **VARY command** has been modified to accept the new FORCE option which allows active remote client sockets to be closed when a LOCAL or REMOTE is varied offline. When FORCE is not specified with OFF (or INACT), the LOCAL or REMOTE is marked offline but any active client sockets are left active. They will be closed at end of session and will then remain closed until the LOCAL or REMOTE is varied back online. Please refer to the Console Subsystem guide for additional information and command syntax.

  Note: This logic change was introduced into 230 under APAR 2300156.

- The **DPARM command** has been modified to accept the new MODIFIERS option which restricts the DPARM display to command modifiers only. This provides symmetry with DPARM EXEC command which restricts the DPARM display to HNAS start parameters only. DPARM ALL (or DPARM with no arguments) will continue to display both command modifiers and start parameters. In addition, when modifiers are displayed, their accepted values will also be displayed.

  Note: This logic change was introduced into 230 under APAR 2300161.

- The **HELP command** text has been moved from the CONSHELP module to each command processor module so that when a change is made for a command, the help text can also be changed in the same module without affecting CONSHELP.

  Note: This logic change was introduced into 230 under APAR 2300161.

- The **DPARM EXEC** command has been modified to display ALLON|ALLOFF instead of ON|OFF for TRCBFR, TRCDATA, TRCDISP, TRCIO, TRCLU, TRCMCH, TRCMCHX and TRCVC to more properly describe *global* trace states

- The **PRNT QLLC command** (and start parameter **PRNTQLLC**) were added so that NAS8xxxx QLLC alert messages could be filtered independently from PRNTVC NAS5xxxx and NASAxxxx VC alarm messages.

  Note: This logic change was introduced into 230 under APAR 2300167.

- The **DNAS command** has been modified to display the Host OS and Version information and the execution start date and time. Please refer to the Chapter 5 migration section for a complete description of the display output changes.

Note: This logic change was introduced into 230 under APAR 2300168.

QLLC Enhancements:

- **QLLC Callout via TYPE=SPU REMOTE**

HNAS will initiate an SVC connection to an SPU that is defined by a TYPE=SPU REMOTE definition statement when OPTIONS=CLOTINITYP=BIND|TIMER|CONSOLE is specified. The DTEADDR=, DCEADDR=, FAC= and CUD= operands on the same TYPE=SPU REMOTE definition statement provide, respectively, the called DTE address, calling DTE address, facilities and call user data for the outbound Call Request packet.

For OPTIONS=CLOTINITYP=BIND, an Outbound QLLC call is initiated when any SLU defined in the LUNAME= operand for the SPU is bound (acquired).

For OPTIONS=CLOTINITYP=TIMER, an Outbound QLLC call is initiated automatically when the OPTIONS=MCHTMR=*value* expires for the *first* defined MCH.

For OPTIONS=CLOTINITYP=CONSOLE, an Outbound QLLC call is initiated when the VARY ACT console command is entered for the named SPU.

For all CLOTINITYP values, an established call remains active until cleared by the remote SPU.

OPTIONS=CLOTFAILRTYLMT=*value* may also be specified to limit the number of callout retries in the event of a call failure. A default *value* of 3 is assumed if omitted. If a zero *value* is specified, failed calls will be retried indefinitely until a connection is established.

OPTIONS=CLOTCONLMT=*value* may also be specified to limit the number of callout attempts after a successful connection has been cleared. A default *value* of 0 is assumed if omitted. If a zero *value* is specified or is set by default, HNAS will attempt to reestablish the connection to the SPU after a forced delay based on the CLOTINITYP setting. This operand is provided to limit the use of X.25 network facilities for an SPU. When the number of call attempts reaches the CLOTCONLMT *value*, a subsequent call can only be initiated via the VARY ACT console command.

- **QLLC Callout via TYPE=MCH REMOTE**

HNAS will initiate an SVC connection to all SPUs that are defined in an SVC3= operand on a TYPE=MCH REMOTE definition statement if they are associated with a callout DTE address (O specified as DTE address delimiter). If the HNAS call completes normally, the connection remains in affect until cleared by the remote SPU. Following a clear, HNAS will attempt to reestablish the connection to the SPU after a forced delay.

Outbound QLLC calls are initiated automatically when the OPTIONS=MCHTMR=*value* expires for an MCH. Initiation of outbound QLLC calls will continue until a Call Accept packet is received. An accepted call remains active until cleared by the remote SPU.

- **QLLC Callout Summary:**

  - **QLLC Callout via Application Bind**

    HNAS will initiate a QLLC SVC connection for an SPU when an Application Bind is received on *any* one of the SLUs identified in the SPU's LUNAME= operand if, and only if, OPTIONS=CLOTINITYP=BIND is specified for the SPU. PLU BINDs that are received before the SPU is connected are queued for delivery after the SPU and its SLUs come active. After the SPU connection is established, an XID request/response exchange is performed followed by an ACTPU and ACTLUs for all defined SLUs in the LUNAME= operand. Any BINDs that have been queued for transfer are then passed to the target SLUs. In the case of a queued BIND, no SLU input and hence no REQSESS is required to establish a SLU/PLU session. The queued BIND already does this. If the SPU is already connected when a BIND is received, the BIND is treated in the normal fashion and is immediately passed to the remote SLU.

  - **QLLC Callout via Timer Control**

    HNAS can initiate a QLLC SVC connection based on a timer. There are two (2) methods for timer initiated callout.

    1) If OPTIONS=CLOTINITYP=TIMER is specified for an SPU, a callout is initiated when the MCHTMR= value expires for the *first* MCH defined in the CDF. This MCH does not have to be associated with any SPU. It is used solely to provide an event that initiates SPU callout processing.

    2) If an SPU is identified in the SVC3= operand for an MCH and the SVC3= entry has a callout DTE address (identified by the character 'O' following the DTE address), HNAS will initiate a connection for the SPU when the MCHTMR= value expires for *this* MCH.

    Note: OPTIONS=CLOTINITYP=TIMER callout initiation takes precedence over SVC3= callout initiation. That is, HNAS will look for all SPU with OPTIONS=CLOTINITYP=TIMER before it looks for any identified in an SVC3= operand.

  - **QLLC Callout via HNAS Console Control**

    HNAS will initiate an QLLC SVC connection for an SPU when the following HNAS Console Command is executed if OPTIONS=CLOTINITYP=CONSOLE|TIMER|BIND is specified:

    RNM=*spuname* VARY ACT

    In the case of OPTIONS=CLOTINITYP=CONSOLE, this command is the *only* way to initiate an SPU callout.

    In the case of OPTIONS=CLOTINITYP=TIMER|BIND, this command is the *only* way to initiate an SPU callout after the call connection limit (OPTIONS=CLOTCONLMT=*value*) has been reached.

- **QLLC TYPE=SPU LUNAME= SLUname MXTname association (**applid and mxtname support**)**

(LUNAME= operand enhancement for QLLC SPUs (SLU *applid* and *mxtname* support)

An APPLNAME= index and MXT name may now be associated with the QLLC SLUs identified in the LUNAME= operand on a TYPE=SPU REMOTE definition statement. The ***applid*** and ***mxtname*** for LLC3 SLUs provide the same function as they do for LLC0 and LLC5 SLUs (via the SVC0= and SVC5= operands). The ***applid*** allows an SLU to be 'wired' (dedicated) to a specific host application. The ***mxtname*** is used to override operands that are specified on the root REMOTE statement and, for the FASTRUN process, to supply overriding VTAM parameters that are propagated to the generated APPL statement for the SLU in the HNAS AMNF. The LUNAME operand for a TYPE=SPU REMOTE definition statement now has the following form:

LUNAME=(...,*sluname* / *rpcecnt* / *spcecnt* / *applid* / *mxtname*,...)

sluname is the SLU name (no change).
rpcecnt is a receive pacing count value (no change).
spcecnt is a send pacing count value (no change).
applid is an APPLNAME= operand index value (new).
mxtname is a TYPE=MXT REMOTE name (new).

Note that the MRMT console command process has been modified to allow all suboperands of the LUNAME= operand on a TYPE= REMOTE to be added, changed or deleted.

- **QLLC SPU IDLETO= Inactivity Clear Support.**

You may now specify the IDLETO=*minutes* operand on a TYPE=SPU REMOTE definition statement to force HNAS to clear an SPU call if no VC or LU activity occurs within the *minutes* interval. This operand is provided to limit the use of X.25 network facilities for an idle SPU.

- **QLLC SPU LUNAME= LOCADDR Coding Improvements.**

The **LUNAME=** operand that is coded for a TYPE=SPU REMOTE definition statement may now contain a 'gap count' rather than consecutive commas to separate SLU names and their related parameters when the SLU LOCADDR values are non-sequential. For example:

coding

```
                                      1
           1      2      3456789      0 <- LOCADDR values
   LUNAME=(,slunm2,slunm3,,,,,,,slunm10)
```

is equivalent to coding

```
                                  1
           1      2      34-9     0     <- LOCADDR values
   LUNAME=(,slunm2,slunm3,6,slunm10)
```

This logic was added to HNAS 230 APAR 2300033 on 05-13-2004.

- **QLLC SPU APPLNAME=ACQUIRE Printer Bind Support Improvement.**

The **APPLNAME=** operand on a TYPE=SPU REMOTE definition statement has been modified to accept ACQUIRE as a keyword (reserved) value. When this APPLNAME= operand entry is selected via a ‘hard wired’ *applid* value (see LUNAME= operand) or via USSTAB processing, the ACB for an SLU is opened and a SETLOGON request is passed to VTAM but a REQSESS request is withheld. This operation conditions the SLU to accept an application BIND but does not solicit the BIND. The SLU remains passive. This support is primarily used for printer LUs that are acquired based on need via requests from other LUs.

Note: This new feature was introduced into 230 as Enhancement APAR 2300030.

- **QLLC SPU OPTIONS=REUSEBUSYSPU support added.**

The **OPTIONS=** operand on a TYPE=SPU REMOTE definition statement has been modified to accept REUSEBUSYSPU as a suboperand value. This new operand allows the SPU to be allocated to a new call when it is already connected. This may be required if the QLLC PAD fails to notify HNAS of an SPU disconnect. When the REUSEBUSYSPU option is effect, the old VC call is cleared and the new VC call is allowed to proceed. When the REUSEBUSYSPU option is not in effect, the new VC call is cleared and the old VC call is allowed to continue. In the case, the following alarm message is generated:

NAS8102W PU FOR DTEADDR=DDDDDDDDDDDDDDD IDBLK/IDNUM=*xxxxxxxx*
ON MCH *mchname*
NAS8102W WAS IN USE WHEN SELECTED

Note: This new feature was introduced into 230 as APAR 2300116.

Enhancements - Expanded Content:

- Console command **MRMT** was enhanced to allow users to modify QLLC (LLC3) SLU definitions in the LUNAME= operand of TYPE=SPU REMOTE definition statements via privileged local or remote console access as follows:

MRMT RNM=*spuname* LUNAME=(*locaddr*,
{*sluname*i|*}
[/{*rcvpac*i|*}
[/{*sndpac*i|*}
[/{*applid*i|*}
[/{*mxtname*i|*}],...
:
{*sluname*n|*}
[/{*rcvpac*n|*}
[/{*sndpac*n|*}
[/{*applid*n|*}
[/{*mxtname*n|*}])

Where: ***locaddr*** represents the staring position in the LUNAME= operand list where SLU name replacement, addition or deletion will start. The number of ***sluname*i** that follow the ***locaddr*** value cannot exceed the extent of LUNAME= operand list which is current set to a maximum of 255 entries.

To delete an SLU entry in the LUNAME= operand list, enter an asterisk in place of the SLU name.

To change an SLU name in the LUNAME= operand list, enter a new SLU name for the existing ***locaddr*** entry.

To add an SLU name to the LUNAME= operand list, enter a new SLU name for a new ***locaddr*** entry. Note that adding SLUs to a LUNAME= operand on any TYPE=SPU REMOTE definition statement requires that HNAS be configured for dynamic SLU support. This is done by creating a dynamic SLU pool using the **OPTIONS=LUDRPOOLCNT=*count*** operand on the BUILD definition statement. You can reserve up to 65534 dynamic SLUs for this purpose depending on the amount of available memory. Each SLU takes approximately X'638' bytes.

In addition to changing or adding an SLU name, you may also change, add or delete(*) the receive pacing count (***rcvpac*i**), send pacing count (***sndpac*i**), the APPLNAME= index (***applid*i**) and/or the the name of a TYPE=MXT REMOTE (***mxtname*i**) for an LUNAME= operand entry. If you wish to modify ***rcvpac*i**, ***sndpac*i**, ***applid*i** or ***mxtname*i** without changing the ***sluname*i** for a ***locaddr*** entry, omit the ***sluname*i** value (***/*** immediately follows ***locaddr***) and specify the other suboperands as follows:

MRMT LUNAME=(*locadrdr*,/*rcvpac*i/*sndpac*i/*applid*i/*mxtname*i,...).

Note: The MRMT command **will not** change or delete an existing LUNAME= operand entry if the SLU is not idle. To make an active SLU idle, you must issue the VTAM **V NET,INACT,*sluname*i** command. An SLU on an SPU that does not have a VC connection is, by default, idle.

The MRMT command processes SLU names left to right in the LUNAME= operand list. If an error occurs, it is reported and subsequent command processing is aborted. This means that it is possible for the MRMT to make some but not all of the changes you requested. To see all the changes that were made before the error occurred, issue the **DRMT LUNAME=** command after the MRMT command.

Examples:

Assume that for **RNM=*spuname* DRMT LUNAME=** the following display is presented:

```
RMTNAME  OPERAND   TYPE
spuname  008AB254  SPU
         LUNAME=********/   /   /  /
                Q1990102/   /   /  /
                Q1990103/   /   /  /
```

This SPU has SLUs defined at LOCADDR=2 and 3. To add an SLU for LOCADDR=5, enter **MRMT LUNAME=(5,Q1990105)** (the SLU name(s) you enter are your choice). Following this command, enter DRMT LUNAME= which will produce the following display:

```
RMTNAME  OPERAND   TYPE
spuname  008AB254  SPU
         LUNAME=********/   /   /  /
                Q1990102/   /   /  /
                Q1990103/   /   /  /
                ********/   /   /  /
                Q1990105/   /   /  /
```

To delete the SLU for LOCADDR=3, enter **MRMT LUNAME=(3,*)**. Following this command, enter DRMT LUNAME= which will produce the following display

```
RMTNAME  OPERAND   TYPE
spuname  008AB254  SPU
         LUNAME=********/   /   /  /
                Q1990102/   /   /  /
                ********/   /   /  /
                ********/   /   /  /
                Q1990105/   /   /  /
```

To add an SLU for LOCADDR=1 and 4, enter **MRMT LUNAME=(1,Q1990101,,,Q1990104)**. Following this command, enter DRMT LUNAME= which will produce the following display:

```
RMTNAME  OPERAND   TYPE
spuname  008AB254  SPU
         LUNAME=Q1990101/   /   /  /
                Q1990102/   /   /  /
                ********/   /   /  /
                Q1990104/   /   /  /
                Q1990105/   /   /  /
```

To add pacing counts for LOCADDR=1, 2 and 3 while also adding a new SLU name for LOCADDR=3, enter **MRMT LUNAME=(1,/2/3,/2/3,Q1990103/2/3)**. Following this command, enter DRMT LUNAME= which will produce the following display:

```
RMTNAME  OPERAND   TYPE
spuname  008AB254  SPU
         LUNAME=Q1990101/002/003/  /
                Q1990102/002/003/  /
                Q1990103/002/003/  /
                Q1990104/   /   /  /
                Q1990105/   /   /  /
```

To delete SLUs for LOCADDR=4 and 5, enter **MRMT LUNAME=(4,*,*)**. Following this command, enter DRMT LUNAME= which will produce the following display:

```
RMTNAME  OPERAND   TYPE
spuname  008AB254  SPU
         LUNAME=Q1990101/002/003/  /
                Q1990102/002/003/  /
                Q1990103/002/003/  /
```

- The HNAS **PING command** has been added to the console subsystem that will allow the console operator to check on the existence of a router on the IP network as well as a specific MCH on the router. The HNAS PING command is not a traditional IP PING command but rather an X.25 PING. The HNAS PING command requires an IP address to identify a router and an X.25 DTE address to identify the MCH link. These two pieces of information allow HNAS to establish an IP connection to the router and then an X.25 (XOT) connection via the MCH link. An indication of success or failure for the PING request is reported back to the console operator.

The format of the PING command is as follows:

```
[IPADDR=a.b.c.d]
[CLGADDR=ddd.ddd]
[CLDADDR=ddd...ddd] PING [XTP|XOT] a.b.c.d ddd...ddd [ifn] [vcn]
```

If you set the IPADDR= and CLDADDR= modifiers, they become the default IP address and called DTE address for the PING command. They will be used if PING is entered with no arguments. If PING is entered with no arguments and IPADDR= and/or CLDADDR= is not set, the command will be rejected. XOT is the default REMOTE type unless you specifically enter XTP. You can always override the default IP address and/or called DTE address by specifically entering them as PING command arguments. If you set the default IP address and only want to change the called DTE address, enter **PING , *ddd...ddd***

For XTP, 2 additional arguments are required: ***ifn*** selects the actual MCH on the IBM router and ***vcn*** selects the LCN 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 canned as follows:

facilities: 080100420707430202 cud: 01000000HNASPING

The target IP address selects the router. The target DTE address selects the MCH serial interface on the router. Note that the router must be configured to map the target DTE address to a specific serial interface.

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. The Clear cause and diagnostic codes can then be used to determine how far the actual calling function has gone. The Cisco doc and HNAS doc lists most clear reason and diagnostic codes.

- **TRCTRAP= diagnostic automation tools**

TRCTRAP= support provides automation tools for improved debugging and trace suspension.

Sometimes it is difficult to capture the cause of a problem in the HNAS internal trace table when tracing has to be stopped manually. In many cases, the only way to resolve a problem is to run HNAS traces. If a problem is intermittent, TRCPRNT 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 may be difficult and, in some cases, may even be impossible.

HNAS has been modified to accept a new configuration operand and console command that will provide the ability to suspend internal tracing (and TRCPRNT) based on alarm message ID match, 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 the trace and TRCPRNT activity is suspended (NAS0050A message generated), HNAS must be shutdown with QY/*password* so that the internal trace table can be formatted. Because the trace is suspended (frozen), HNAS need not be shutdown immediately but can be shutdown at your convenience. If the trap was performed with

TRCPRNT enabled, the SYSPRINT file can be manually switched (see PRNT CLOSE|OPEN *dsname*|CLSOPN dsname console 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). Note: A second phase of this enhancement will be provided which will locate the trace data into a separate SYSOUT or DSN= file so that a shutdown is not required. The following text identifies the new HNAS BUILD TRCTRAP configuration operand and console command that may be used for trace trap processing.

CONFIGURATION ENHANCEMENT:

```
BUILD TRCTRAP=(ALRMLIST=(id1,...,idn),
      RCVLIST=(prot1||pkt1,...protn||pktn),
      XMTLIST=(prot1||pkt1,...protn||pktn))
```

CONSOLE COMMAND ENHANCEMENT:

```
TRCTRAP {ALRMLIST=?|*|(id1,...,idn)}
        {RCVLIST=?|*|(prot1||pkt1,...protn||pktn)}
        {XMTLIST=?|*|(prot1||pkt1,...protn||pktm}}
        {SHOW} {SUSP} {RSME}
```

FOR BOTH CONFIGURATION AND CONSOLE COMMAND:

idi => alarm ID (e.g, NAS1001I) that may contain asterisks (*) as wildcards (e.g., NAS1****). A maximum of 16 idi entries may be specified.

proti => packet protocol (00=>XTP, 04=>XOT).

pkti => packet data from PKTTYPE for length *m* that can contain all ones as wild cards (e.g., 13FFFF).

A maximum of 512 bytes for protocol and packet data may be specified which include a length byte (m) for each packet. The packet data is compared only for the length (m) that is given. 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.

Note: A match on any value in the ALRMLIST=, RCVLIST= or XMTLIST= operands can suspend tracing. The first match in any of these operands is the winner. Multiple operands with multiple values are provided to allow for the greatest flexibility.

FOR CONSOLE COMMAND ONLY:

Enter: * as first operand value to reset operand
? as first operand value to display operand
SHOW to display all operand values
SUSP to suspend trace manually
RSME to resume suspended trace

In addition to the list display produced by TRCTRAP ALRMLIST=? or TRCTRAP RCVLIST=? or TRCTRAP XMTLIST=? 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
```

Note: When values are specified for the ALRMLIST=, RCVLIST= and XMTLIST= command arguments, they are appended to the end of the corresponding table in the order in which the values are entered. For example:

If `BUILD TRCTRAP=(ALRMLIST=(NAS8***I)` was specified in the CDF and if
`TRCTRAP ALRMLIST=(NAS2021I,NAS2100I)` then
`TRCTRAP ALRMLIST=(NAS2201I)` are issued as console commands, the
`TRCTRAP ALRMLIST=?` console command would produce the following display:

```
ALRMLIST=NAS8***I
         NAS2021I
         NAS2100I
         NAS2201I
```

CONFIGURATION EXAMPLES:

Trap multiple events:

```
BUILD 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
```

Trap PAD logon request:

```
BUILD TRCTRAP=(RCVLIST=(04FF6CA063F0,  <- l cp (even)
                        04FFEC20E370,  <- l cp (odd)
                        04FF6C206370,  <- l cp (none)
                        04FFCCA0C350,  <- L CP (even)
                        04FF4C2043D0,  <- L CP (odd)
                        04FF4C204350)) <- L CP (none)
```

Trap QLLC Terminate-Self PIU:

```
BUILD TRCTRAP=(RCVLIST=(04FF2C0000FFFFFF0B8000010683)) <- TERM-SELF
```

TRCTRAP ALERT MESSAGES:

The following alerts are provided when tracing is suspended due to an alarm ID or packet match.

```
NAS0050A TRACING SUSPENDED DUE TO TRAPPED ALARM ddddddd
NAS0050A TRACING SUSPENDED DUE TO TRAPPED INPUT PACKET xx...xx
NAS0050A TRACING SUSPENDED DUE TO TRAPPED OUTPUT PACKET xx...xx
```

The following alert is provided when tracing is suspended via the `TRCTRAP SUSP` or `TRCALL SUSP` command.

```
NAS0050A TRACING SUSPENDED BY COMMAND
```

The following alert is provided when tracing is resumed via the `TRCTRAP RSME` or `TRCALL RSME` command.

```
NAS0060W TRACING RESUMED BY COMMAND
```

The following alert is provided after the NAS0050A message (for `TRCTRAP SUSP`) or NAS0060A message (for `TRCTRAP RSME`) when trace trap filtering is active. The message is also issued if trace trap filtering is enabled via the `TRCTRAP` configuration operand.

```
NAS0070W TRACE TRAP FILTERING ACTIVE
```

The following alert is provided after the NAS0050A message (for `TRCTRAP SUSP`) or NAS0060A message (for `TRCTRAP RSME`) when trace trap filtering is inactive.

```
NAS0070W TRACE TRAP FILTERING INACTIVE
```

GENERAL NOTES (SUMMARY):

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 MINDATA` or `TRCVC MAXDATA` should be active.

4) 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 arguments rather than giving them a custom trap modification. This summary intended to provide some information about the TRCTRAP function. We do not, however, expect customers to use TRCTRAP functions without Comm-Pro involvement.

Note: This new feature was introduced into 230 as Enhancement APAR 2300045/2300047.

HNAS V2R2M0 New Features

V2R2M0 General Availability - 11-11-2002

- All appropriate V2R1Mn **APARs** incorporated as permanent fixes into this release. Please refer to the appropriate APAR Summaries for additional information.

- **High Memory Support** was added to this release to reduce low memory requirements and allow substantially larger configurations. The feature is enabled when **APFXEQ** is coded on the PARM= start parameter. This new support allows HNAS to allocate free memory from above the low memory line which is typically 16MB. You may specify up to 7 sub-pools using the APFMEMSP= operand. For example: **APFMEMSP**=(229,230). Memory is allocated in the sub-pool from left to right, so that for this example memory in sub-pool 230 will not be used until all memory in sub-pool 229 is exhausted.

- **Multiple** 'same type' **Server** enhancement has been added to V2R2M0. It allows the use of the same or different IP addresses for the same type of LOCAL as well as a socket limit of up to 65535 (the theoretical limit) per LOCAL. When more that 2000 sockets are specified per LOCAL, you need to specify the SHAREPORT operand in the TCPIP PROFILE so that the stack will allow port 1998 to be used for more than 2000 connections. Within HNAS, each set of 2000 sockets is treated internally as a server. So if you code **SOCLMT**=6000 on a LOCAL statement, this will create 3 servers for the same IP address and port number. The stack will present connections to the 3 servers in a round robin fashion. With this feature there is no longer a requirement to run multiple copies of HNAS if you need to support more than 2000 sockets.

- **Multiple Stack Support** enhancement has been added to V2R2M0. This allows the same HNAS address space to act as server for more than one TCP/IP Stack. The TCPNAME operand which in the past was only allowed on the BUILD definition statement may now also be coded on the LOCAL definition statement to identify a specific stack for the server.

- **Shared Socket Support (REMOTE TYPE=XOT IPADDR=*a.b.c.d*,PORT=1998)**. For Cisco routers, this feature eliminates the need to pre-define separate pools of inbound and outbound TCPIP sockets for a specific router in the HNAS configuration. Specifying IPADDR=*a.b.c.d*, PORT=1998 on a TYPE=XOT REMOTE definition statement (which identifies a specific router) creates a single pool of TCPIP sockets that can be used for both inbound and outbound sessions. The number of sockets in the pool is determined by the VCLMT= operand. A socket from the pool is allocated to a activating session on a first come, first served basis. The same socket can be used for an inbound initiated connection at one time then an outbound initiated connection at another time. Please refer to the Chapter 4 operand IPADDR= for a complete description of this support.

- **Authorization Key Support** is now required in V2R2M0. HNAS **now requires an authorization file (NASAUTH)** in order for it to operate beyond the 90 day trial period. The authorization file is specified by the **//AUTH DD DSN=QQQQ.HNASMAC(NASAUTH),DISP=SHR** statement in the HNAS start job.

  For users undergoing HNAS trials, a NASAUTH file is provided with a 90 day expiration date. Customers who have completed their trial tests and have purchased HNAS will be given a permanent authorization file that will never expire. Existing customers upgrading to 220 from an earlier release will be provided with a permanent key when they acquire the upgrade.

  Please refer to the HNAS Authorization Considerations content under the Chapter 2 Activation and Operation section of the HNAS Guide and Reference manual for additional information.

- **Qualified Logical Link Control** (**QLLC**/LLC3) over XOT support has been added. Native SNA support for Type 1 PUs (maximum of 31 SLUs per PU) and Type 2 PUs maximum of 255 SLUs per PU) and LU Type 0, 1, 2 and 3 device support over QLLC (PUT2.1 or LU 6.2 not currently available). This support is new for V2R2M0 and eliminates the previous requirement to employ DLSW for QLLC line attachment. For additional information please see QLLC operand and parameter additions at the end of this document.

- **A**pplication **M**ajor **N**ode **F**ile generation (**AMNF**) support has been added. When PARM=FASTRUN is specified and the configuration return code is 4 or less, HNAS will produce an AMNF using the SLU names that are specified in the **LUNAME**, **PVC**, **SVC0**, **SVC4** and **SVC5** operands from all REMOTE definition statements in the **C**onfiguration **D**ata **F**ile (**CDF**). The AMNF is placed in a dataset identified by the MAJNODE DD statement. The MAJNODE dataset can be a sequential file or a member of a partitioned dataset. DISP=OLD is required. Automatic AMNF generation ensures that there will never be a mismatch between the SLUs specified in the CDF and those specified in the AMNF. Application specific VTAM parameters can now be coded on the respective CDF resources. The VTAM parameters will be promoted along with the SLU names when using the AMNF generation process.

- **Multiple SYSOUT Support** has been added to this release. This allows the local or remote console operator to Close and Open (switch) the active SYSPRINT DSN=dsname to any entry defined in the HNASXEQ JOB/JCL.

- The HNAS **Timer Queue** logic was improved to reduce processing cycles for environments with large configurations (LUs, VCs and REMOTE resource definitions).

- Various **CDF Options** and **Parameters** were added or changed to allow the user greater flexibility when defining resources in V2R2M0:

  - **Callout Alternate DTE Connect Retry Support -**
    SVC0=(...,lu-name/dteaddr1-dteaddr2-dteaddr3O/mxt-name,...) may be coded to enable callout connection retry using up to 2 alternate called DTE addresses across TYPE=XOT REMOTEs identified by the RTEOUT= list on a LOCAL TYPE=XOT statement. If the call to dteaddr1 fails (CLEAR or timeout response to HNAS Call Request packet), then HNAS will try the call using dteaddr2 and dteaddr3 (if necessary). This logic is available for LLC0 and LLC5 callout operation (SVC0= and SVC5=).

  - **Callout Session Connect Balancing -**
    OPTIONS=BALANCERTEOUT operand has been added to the **TYPE=XOT** LOCAL definitions to enable Round Robin connection balancing across TYPE=XOT REMOTEs identified in the **RTEOUT=** list. HNAS now has the ability to provide a form of load balancing for host initiated (callout) sessions across multiple routers.

  - **Callout Connection Routing using Calling or Called Address -**
    RTEOUT=(remote-name1/dteaddr1T,remote-name2/dteaddr2S,..) The RTEOUT= operand of the TYPE=XOT LOCAL statement has been modified to direct routing of HNAS call requests by called or calling address. When HNAS places a call the called and calling addresses are known. For LLC0 and LLC5 the called address comes from the SVC0= or SVC5= operands on the TYPE=MCH REMOTE statement. The Calling address comes from the DCEADDR= operand. For GATE (LLC4) the addresses are provided by the CTCP. ‘T’ in the RTEOUT= operand specifies that the called address (target) is to be used for RTEOUT= entry selection. ‘S’ specifies that the calling address is to be used for entry selection. Note that ‘T’ is the default and provides downward compatibility with earlier releases.

- **Callin DTE Address Filtering -**
  LLC0 (SVC0=) and LLC5 (SVC5=) Callin DTE Address filtering for **TYPE=MCH|XTP** REMOTEs SLUname association. HNAS now supports filtering of DTE addressed based on the number of digits coded on the SVC0= and SVC5= dteaddri suboperands. In earlier releases the inbound DTE address in the call request packet had to match the SVC0|5 dteaddri value exactly for LU selection to occur. Please refer to Chapter 4 for revised information on SVC0 and SVC5 coding rules.

- **Callin SLU/PLU Fixed Connection Support -**
  SLU/PLU fixed connection support for **PCNE LLC0 (SVC0)** and **PAD LLC5 (SVC5)** now supports coding of an MCH **APPLNAME=** index entry for direct mapping instead of the previous 211 requirement for a SYSL=DATA=char system select value. Please refer to Chapter 4 for revised information on SVC0 and SVC5 coding rules. The following table depicts the coding differences between 211 and the new 220 coding requirements:

Old Method Under V2R1M1	New Method Under V2R2M0

`SVC0=(2,`
`MCH10001/123456IA,`
`MCH10002/X001540IB)`
`SYSL=(DATA=A/0,DATA=B/1)`
`APPLNAME=(TSO,CICS)`
	`SVC0=(2,`
	`MCH10001/123456I0,`
	`MCH10002/X001540I1)`
	`APPLNAME=(TSO,CICS)`

- **Callin Default PLU Support (SYSL= now Optional) -**
  Callin Default PLU assignment via SVC0=/SVC5= APPLNAME= association is now supported in HNAS. For **LLC0 and LLC5 inbound calls**, after HNAS sets the LLC TYPE and locates an LU for the call, a PLUname for the session must be determined. If the name is not provided by the **SVC0/5=** operand (SVC5=(...,LU1/I2,...) where 2 is an index in APPLNAME=) or by SYSL= then the first **APPLNAME=** entry is used as a default. See sample coding above. This default is also taken if **SYSL=** is omitted (**SYSL= was required** in previous releases and is **now optional**). Please refer to Chapter 4 for revised information on SVC0= and SVC5= coding rules.

- **Callin GATEFC REQSESS Support**
  Callin GATEFC logic now supports an option which causes HNAS to issue a REQSESS for each terminal session SLU to force an acquire from the target PLU. This is done to ensure that the terminal SLUs are in the active state (BIND/SDT) to accept incoming calls. This new logic is enabled when an asterisk (*) is appended to the end of a plunamei value (e.g., CSFI*) on the REMOTE LUNAME= operand.<Ref: 11-20-2002>

- **OPTIONS=LCN0USED** may be specified on the **TYPE=MCH** REMOTE definition to force HNAS to start the GATE resource identifier (RESID) at zero rather than at one (plus the PVC count). **GATE=GENERAL,PVC=NONE** is also required. Some **CTCPs** (**CSFI** for example) know when LCN0 is used on an MCH so they expect the first RESID to be zero not one. Without this option a CTCP, which is configured for LCN0=USED, would think that the second SLU in the SVC4 operand was being used rather than the first. OPTIONS=LCN0USED ensures that HNAS and the CTCP are in synch.

- **OPTIONS=MCHTMR=value** operand has been added to the **TYPE=XOT/XTP** REMOTE definitions to allow users to fine tune the activation delay for GATE Control Session and Fast Connect

LU Sessions. This timer also applies to activation of the SLUs defined for callout support. In previous releases of HNAS the timer value was fixed at 60 seconds which is also the new parameters default value. <Ref: 210C002 05-10-2002>

- **OPTIONS=TCPRBLMT=value** operand was added to **TYPE=XOT/XTP** REMOTE definitions to allow users to reduce the number of staging buffers associated with each TCP/IP socket (XOT Virtual Circuit VC). In previous HNAS releases the buffer count was fixed at 7. We have observed that a value of 2 is adequate for a variety of transaction processes as long as the buffer will accommodate a complete XOT packet. Configuration environments with hundreds of sockets/VCs will benefit from the reduced memory requirements when running under the reduced buffer count. The new parameter default buffer limit value currently remains at 7 for XTP and is now 2 for XOT.

- **OPTIONS=LUBLTCNT=value** operand has been added to the BUILD definition to allow users to control the length of the Buffer List Table. The table is used to process all data packets in an m-bit chain for transfer to the application. In V2R1Mn the length was increased to 40 while earlier releases had a length of 30. <Ref: 2100024 06-22-2002>

- **HOME=value** operand has been added to the **TYPE=XOT|XTP** REMOTE definition to allow users to associate a REMOTE with a LOCAL resource. This new operand operates in conjunction with the Multiple Server Support. If a HOME values isn't specified a default will be assigned at activation.

- **SOCLMT=value** operand has been added to the **TYPE=XOT|XTP** LOCAL definition to allow users to control the number of sockets available under each LOCAL definition. The previous limitation restricted users to 2000 sockets per HNAS address space. This new operand is used in conjunction with the Multiple Server Support.

- **TAP=0** is the new default operand for **TYPE=XOT|XTP** REMOTE definitions. In previous releases the default TAP (keep-alive) value was 60. <Ref: 2100015 05-16-2002> The revised logic also restricts TAP= value parameter coding to the first **TYPE=XOT|XTP** REMOTE definition per unique IPADDR. In previous releases the CDF decode would allow TAP=value coding on all remotes even though only the TAP= parameter from the primary (first) remote was used for the keep-alive timer control.

- **OPTIONS=ECHOFAC** operand has been added to the **TYPE=MCH** REMOTE definition to enable echoing of facilities from the inbound XOT Call Request packet to the HNAS XOT Call Accept packet. Some networks don't support this feature while others can use the response to further negotiate facilities values. A default of **NOECHOFAC** is assumed if no parameter is coded.

- **SUBD=** allowed on **non-Fast Connect GATE MCHs (**CTCP selection using subaddress digits This permits inbound GATE sessions to select a CTCP using subaddress digits in the call request packet. Previous logic allowed SUBD= only for Fast Connect GATE sessions (NPSI restriction). <Added to 220 New Features list on 11-21-2003. See APAR Ref: 2200061 dated 11-21-2003 for additional information.>

- **TAP** keep alive logic was redesigned and now schedules an XOT Call Request instead of an XOT Clear Request to ensure that a protocol level response (in this case a Clear Request) is always received from the router's XOT services. Some XOT router IOS or software levels don't always respond to an XOT Clear Request received on an idle virtual circuit but must respond to a XOT Call Request.

TAP= logic can now perform protocol level tapping with XOT Call or XOT Clear packet. See new parameter OPTION=TAPWITHCLR that was added on 2003-08-27 via APARs 2200048 and 2200052.

- The **FASTRUN** process will now allow the name for the AMNF VBUILD statement to be specified using the APPLNAME= operand rather than NASNAME= operand on BUILD definition statement (APAR 2200080).

- The **FASTRUN** start parameter option process was updated to produce a **Memory Requirements Summary Report**. This new feature is extremely useful in identifying storage requirement for various HNAS system components such as: program size, buffer, trace tables and a variety of control block types.

- **PRNTDATE** start parameter option is now supported which allows users to enable a special DATE prefix in front of the TIME prefix for SYSPRINT log entries. The **yy.ddd** format provides a 2 digit year, a period and the Julian date. This will improve SYSPRINT log viewing capabilities.

- **Several enhancements** were made in this release to improve **Alarm Event** and **Alert Message** delivery processes and provide alert event counter history:

  - **ALRMLMTS** operand has been added to the BUILD definition statement to allow alarm limits and the alarm table size to be configured (ALRMLMTS=(**T**ime_interval, **I**nformational_cnt, **D**efault_cnt, **A**ction_cnt, **W**arning_cnt, **E**rror_cnt, **S**evere_cnt, **O**ther_cnt, **L**og_size)) or bypass alarm limits (ALRMLMTS=NONE). The time value and message limit interval values were fixed in our previous release while the alarm logging table size is new to this release:

    - **ALRMLMTS** values may be modified via the **ALARM** console command (ALARM LIMITS=(**T**ime, **I**cnt, **D**cnt, **A**cnt, **W**cnt, **E**cnt, **S**cnt, **O**cnt)) or disabled using the console command (ALARM LIMITS=NONE) which resets the settings to zero's effectively disabling the limits. The **alarm logging table** count may not be modified once HNAS is running, it remains fixed by the value specified in the BUILD ALRMLMTS operand.

    - **ALRMLMTS** values may be displayed via the **ALARM** console command (ALARM LIMITS=?).

  - An **alarm logging table** has been added to record alarm ID's and the number of times a particular alarm is issued. The table is ordered by alarm ID. These counters are independent of alarm limits or alarm filtering modes. An alarm logging table is generate for each console PCE (initial local alarm logging limits and table size are based on the ALRMLMTS=values defined in the CDF configuration file:

    - The **alarm logging table** may be cleared (all entries removed) or reset (all counters reset for current entries) via the **ALARM** console command (ALARM LOG=CLEAR or ALARM LOG=RESET).

    - The **alarm logging table** may be displayed to show the alarm IDs and associated counts via the **ALARM** console command (ALARM LOG=?).

  - **ALRMFLTR** (**P**)urge option has been added which allows users to specify whether or not specific alert/event alarm messages should be recorded in the SYSPRINT file. The filtering option available in our previous HNAS release gave users the ability to prevent messages from being displayed at the operator console although they would always be logged in the SYSPRINT file. The major benefit of this new feature allows users to dramatically reduce the size of the

SYSPRINT file by eliminating specific message activity. The alarm counts for all purged events are still recorded in the Alarm Logging Table summary (by type) so that users are aware of specific message activity.

- An **Initialization Complete alert message NAS0001I** is now generated once the initialization phase of the HNAS activation process completes successfully. This new alert message is independent from the HNAS and router connectivity alert messages which were previously used to determine HNAS availability. The new message now provides a more accurate representation of HNAS system availability.

```
NAS0001I HOST NAS INITIALIZATION COMPLETE, ALL FUNCTIONS READY
```

- Following are some of the **alert messages** that were added to convey additional **TCP/IP states**. Please refer to the documentation manual for a complete list of new and reclassified alert message ID’s. <Ref: 2100020 05-31-2002>

```
NAS2321W (TAKESOCKET FAILED)
NAS2311W (GIVESOCKET FAILED)
NAS2301W (GETCLIENTID FAILED)
NAS2268I (ACCEPTED CONNECT PASSED)
```

- **NAS6715W and NAS7715W Alert Message** SYSPRINT entries now include a display of the buffer content for the call request packet that encountered the clear condition. The availability of the call request packet image generally eliminates the previous requirement to run debugging traces to view packet content improving fault isolation resolution times.

•) **NAS2nnns and NAS2nnnI TCPIP Alert Message Enhancement** - The following HNAS TCPIP messages were modified to improve resource identification. <**06-25-2003** - Although this new feature was provided in the initial implementation of 220 we erroneously omitted the description in this section >.

```
  From      To       Message ID
--------  -------- --------------
NAS2nnns->         'cmd FAILED FOR SOCK=a.b.c.d(p) SOCKID=sid RC=rc/erno'
        ->NAS2nnns 'comp=a.b.c.d(p) SOCKID=sid PCEID=pid NAME=compname'
        ->NAS2nnnI 'cmd REQUST FAILED, RC=rc/erno'

Where: cmd = failing TCPIP command.
       comp = failing component (SERVER (LOCAL) or CLIENT (REMOTE)).
       a.b.c.d(p) = IP address and port number for failing component.
       sid = TCPIP internal socket identifier for failing component.
       pid = Process Control Element identifier for failing component.
       compname = name of failing LOCAL or REMOTE component.
       rc/erno = return code and error number for failing TCPIP command.
```

•) **NAS2252E TCPIP Alert Messages** were added to provide additional diagnostic information and event activity for TCP/IP Select processing. <Added to 220 New Features list on 02-19-2004. See APAR Ref: 2200073 dated 02-16-2004 for additional information.>

```
NAS2252E SELECT REQUEST INTERRUPT LOST, RETRY WILL BE ATTEMPTED
NAS2252E SELECT REQUEST INTERRUPT LOST, SOCKET MUST BE CLOSED
NAS2252E CLIENT=iii.iii.iii.iii(port) SOCKID=sockid  PCEID=pceid
                                      NAME=compname
```

  - A **NAME=pce-name** field was added to the TCP/IP SERVER and CLIENT continuation messages to convey the resource name that the process is associated with. The pce-name may refer to a REMOTE, LOCAL, or MCH resource name. This new field provides an immediate association with the alert message and the resources connected with the event.

  - A new default start parameter **SHOWCNFG OFF** was added to eliminate CDF scan messages from being displayed at the master operator console when HNAS is started. Operation's environments that wish to continue seeing these messages can code **SHOWCNFG** on the start parameter to enable the messages. Note that messages are always written to the **SYSPRINT** file unless the **ALRMFLTR** (P)urge option is enabled. This parameter operates independently from the **SHOWOFF** or **SHOWERR** parameters. <Ref: 2100022 06-13-2002> This new release also allows specific start parameter values to contain a space separator character between the primary command and the action value (i.e. SHOWCNFG OFF).

  - Corrective logic for the **SHOWERR** start parameter was developed to prevent HNAS from delivering Information, Default and Comment messages to the master operator console. Only error messages are now delivered to the master operator console in SHOWERR mode (which is the new default). Note that all of these messages will continue to get logged in the **SYSPRINT** file unless the **ALRMFLTR** (**P**)urge option is enabled.

- **Additional trace entries** and the ability to control buffer data information written to the **SYSPRINT** file continue to improve the diagnostic capabilities of this product:

  - **TRCLU** and **TRCVC** sub-operands **MINDATA** (default) **MAXDATA** and **NODATA** can now be coded on the PARM= or set via the HNAS console. The NODATA option dramatically reduces SYSPRINT output when tracing doesn't require partial 'MINDATA' or complete 'MAXDATA' buffer or trace control area data to be displayed along with the trace entries.

  - **TRCBST** start parameter and associated console command has been added to allow tracing of **alarm logging table entry** insertion.

  - **TRCCNFG** start parameter has been added to allow tracing of the entire HNAS configuration process.

- **Additional** HNAS **console capabilities**:

  - Console commands **DLCL and MLCL** were added to allow users to modify **TYPE=XOT|XTP** LOCAL definitions via the privileged local or remote console access.

  - Console command **DPARM** display output was modified to list all of the default start parameter options to improve presentation of the operational parameters.

  - Console modify command logic was added that will allow **sublist** elements to be continued so that you do not have to enter them on one line. The continuation (M-char) is a plus (+) that appears after the record text but before any comments (;). This logic became a requirement as the REMOTE **SVC0=, SVC3= and SVC5=** operand parameter strings continue to expand.

  - Console **DPARM EXEC** argument was added allowing for the displaying of start-up EXEC-PARM= parameters as well as a means of viewing the revised settings after parameter changes are made to the active HNAS environment.
  - Console **HELP** enhancement treats command ? as HELP command.

- Console **VARY** and **DPCE** command has been updated to allow users to enter the actual resource name in the command syntax.

- Console **resource identifier enhancement** allows resource IDs to be entered on either side of a command: (e.g., RNAME=rmtname1 DRMT RNAME=rmtname2). The value on the left side of a command remains the default value for all related commands. The value on the right side of a command overrides the default for the current command only. If a resource ID is entered without a command, it simply sets the default value for all related commands. Please refer to the V2R2M0 documentation manual for a complete list of resource IDs that may be entered on either side of a command.

- **Q**ualified **L**ogical **L**ink **C**ontrol **operand** and **parameter** additions for V2R2M0:

  - **SVC3** and **LLC3** operands have been added to the **TYPE=MCH** REMOTE definition statement. The SVC3 operand has the same form as the SVC0 and SVC5 operands except that instead of specifying SLU names, SVC3 entries are SPU names. SPUs are defined by **TYPE=SPU** REMOTE definition statements.

  - **IDBLK** and **IDNUM** operands have been added to the **TYPE=SPU** REMOTE definition statement. HNAS will allocate an SPU to a QLLC virtual circuit based on an IDBLK and IDNUM match or based on a calling DTE address match from the SVC3 operand. This ensures that a one-to-one mapping is maintained between the logical SPU in HNAS and the real SPU in the network.

  - **LUNAME** operand has been added to the **TYPE=SPU** REMOTE definition statement. The LUNAME operand provides SLU names for the real SPU in the network. The SLU names are positional. The first SLU name is for LOCADDR=1, the second for LOCADDR=2 and so on. If the real SLU local addresses are non-contiguous, the corresponding entries in the LUNAME operand are simply skipped. For example: LUNAME=(SLU1,SLU2,,SLU4,,SLU6) identifies SLUs whose local addresses are 1,2,4 and 6, respectively. Local addresses 3 and 5 are undefined.

  - **APPLNAME** and **SYSL** operands have been added to the **TYPE=SPU** REMOTE definition statement. QLLC SLUs can select a host application via LOGTAB or USSTAB interpretation or based on text specified in the SYSL operand.

  - **CUD, DCEADDR, FAC, LOGTAB** and **USSTAB** operands have been added to the **TYPE=SPU** REMOTE definition statement to override values specified on the root **TYPE=MCH** REMOTE definition statement.

  - **MAXDATA** operand has been added to the **TYPE=SPU** REMOTE definition statement. This allows users to control the MAXDATA value of the SPU level. The default value is 256 (plus the RH count).

  - **OPTIONS** operand has been added to the **TYPE=SPU** REMOTE definition statement. The following connection options are currently supported:

    - **PRIMARY** - marks the SPU as the primary half session in a primary-to-secondary.
    - **SECONDARY** - marks the SPU as the secondary half session in a primary-to-secondary.
    - **PEER** - marks the SPU as a peer half session in a peer-to-peer.

- Additional information regarding product upgrade considerations is available to existing customers in the Migration section of the HNAS Guide and Reference manual.

HNAS V2R1M1 New Features

V2R1M1 General Availability - 06-01-2002

- All appropriate V2R1M0 **APARs** have been incorporated as permanent fixes
- **Z/OS V1R2 (and V1R4)** officially supported in HNAS V2R1M1. While Z/OS V1R1 is supported under HNAS V1R1M4 and V2R1M0, TCP/IP Stack design changes in Z/OS V1R2 required logic changes in HNAS that were initially attempted with APAR correction (210nnnn) and ultimately resolved in the HNAS V2R1M1 release. <Ref: 2100003, 2100011 05-24-2002>
- **SLU Selection by CUD Data -** Allow an SLU to be selected by Call User Data rather than calling DTE address when a Call Request packet is received. <Ref: 210C001 05-07-2002 Item 1>
- **SLU to PLU Connection Wiring** - Allow an SLU connection to a PLU to be 'wired' together without operator/user input. This enhancement permits a user to dedicate an SLU to a specific host application for a PCNE or PAD session by specifying a system select character after the DTE address for an SVC0|5 operand. <Ref: 210C001 05-07-2002 Item 2>
- **OPTIONS=MCHTMR=value** operand has been added to the TYPE=XOT|XTP REMOTE definitions to allow users to fine tune the activation delay for GATE Control Session and Fast Connect LU Sessions. In previous releases of HNAS the timer value was fixed at 60 seconds which is also the new parameters default value. <Ref: 210C002 05-10-2002>

This page left intentionally blank.

HNAS V2R1M0 New Features

V2R1M0 General Availability - 03-01-2002

- All appropriate V1R1M4 **APARs** have been incorporated as permanent fixes.
- The HNAS **dispatcher** has been enhanced to minimize non-productive subtask dispatches which reduces total CPU utilization.
- **Dynamic IP address** assignment (**IPADDR=DYNAMIC**) is now supported for Cisco XOT router environments. This feature eliminates the previous requirement to pre-define routers to HNAS when only inbound connections will be used and reduces the size of the HNAS CDF configuration file and memory requirements. When dynamic IP address assignment is in affect for a router, shoulder tapping (KEEP ALIVE) processing is inhibited.
- **MODIFY** interface made standard and is invoked via the **USEMDFY** start parameter.
- **TRCTASK** console command added to trace subtask WAIT and POST events.
- **TRCSUBR** trace entry expanded to log full register set in HNAS trace table.
- **USSTAB** operand can now be specified on the TYPE=MXT REMOTE definition statements to override USSTAB on the root TYPE=MCH REMOTE.
- **PADPARM** operand can now be specified on the TYPE=MXT REMOTE definition statements to override PAD parameters on the root TYPE=MCH REMOTE.
- **PADPARM**, **CUD** and **FAC** values can now be modified via the **MRMT** console command accepting full list or partial list via offset/value notation.
- **LLC0**,...,**LLC5** entries can now be added or deleted via the **MRMT** console command.
- A **Configuration Error Summary** is now displayed as a function of severity after the CDF is processed. The number of Informational (RC=0 level), Warning (RC=4 level), Error (RC=8 level) and Severe Error (RC=12 level) messages are listed.
- **XOT Call Accept Facilities support**. Facilities values provided in the router's XOT Call Accept packets are now processed by HNAS. This support allows the router to step down the facilities values (like packet and window sizes) originally provided in the HNAS XOT Call Request packet.
- **CONCMDQ** operand has been added to BUILD definition statement to allow HNAS console commands to be executed at HNAS startup without operator intervention. **DNAS** is the default command when the CONCMDQ operand is omitted.
- **ALRMFLTR** operand has been added to BUILD definition statement to allow HNAS console messages to be filtered by message ID.
- **ALRMFLTR** values may be added or deleted via the **ALARM** console Command (ALARM FILTER=(ALLOW|SUPPRESS,id1(A|S),...idn(A|S))).
- **ALRMFLTR** values may be displayed via the **ALARM** console command (ALARM FILTER=?).

- **Alarm limits** are now enforced on all messages written to the system console. This logic was implemented to prevent excessive alarm event activity from inundating the master console log with HNAS Alert messages. Fixed values are as follows:

  - 30 - Time interval (in seconds) in which that limits are in effect.
  - 20 - maximum number of Informational messages allowed during interval.
  - 10 - maximum number of Default messages allowed during interval.
  - 20 - maximum number of User Action messages allowed during interval.
  - 25 - maximum number of Warning messages allowed during interval.
  - 30 - maximum number of Error messages allowed during interval.
  - 30 - maximum number of Severe Error messages allowed during interval.
  - 10 - maximum number of all Other messages allowed during interval.

  - We can provide customized over-ride values for V2R1M0 in the event that our fixed values aren't suitable for your environment. In our upcoming V2R2M0 release the default values can be replaced with user defined values via the revised ALRMLMTS operand.

- Some new **Alert messages** were added to improve environment monitoring capabilities:

  - NAS0910I - Daily Date Cross Over event (issued at midnight) in SYSPRINT for date validation.
  - NAS5000I - Starting Console Session event for remote access monitoring and audit.

- **Remote consoles** can be conditioned to receive HNAS error messages via the **ALARM** console command. Remote consoles, like the system console, are also subject to alarm limits.

- **LOGTAB** operand has been added to the BUILD and REMOTE definition statements to allow LOG table processing in addition to USS table processing (**MCHSOL** must be selected from the APPLNAME operand list). **LOGTAB** value may be modified via the **MRMT** console command.

- **OPTIONS** operand has been added to the REMOTE definition statement to allow activation of special processing. The following options are currently supported:

  - **REPDCEADDR** - causes the calling DTE address in outgoing Call Request packets to be replaced with the value specified for the DCEADDR operand (valid for **GATE=GENERAL** only).

  - **STRIPRTEIN** - causes the called DTE address from incoming Call Request packets to be removed before the packets are passed to the host CTCP application (valid for **GATE=GENERAL** only).

  - **STRIPFAC** - causes the facilities from incoming Call Request packets to be removed before the packets are passed to the host CTCP application (valid for **GATE=GENERAL** only).

  - **OPTIONS** values may be added or deleted via the **MRMT** console command.

- **MONITOR** console command has been modified to display VC and LU connect and disconnect counts at the end of the monitor interval.

- **PACE=0** is no longer necessary to inhibit default messages when pace is omitted for TYPE=MCH resources.

- **Console Prompt** changed from **'COMMAND UNRECOGNIZED, RE-ENTER'** to **'INPUT ERROR, RE-ENTER'** to clearly distinguish from application error messages like those generated by TSO.

Glossary

Glossary of Terms

GLOSSARY

This glossary contains a list of acronyms, abbreviations and terms associated with mainframe (host) communications networking products as related to the IBM 3745 communication controller and IBM NPSI program products. HNAS is a NPSI replacement product primarily utilizing Cisco routers and the XOT (X.25 over TCP/IP) protocol to support remote equipment and host NPSI application with no further requirement for NPSI or 3745 communication controllers.

The descriptions for some of these terms are from the perspective of data communications background with FEPs, Routers and X.25 based equipment and software. Some of the terms and abbreviations are Comm-Pro's interpretation or generalization of usage.

Glossary of Terms

37XX	37XX represents the family of IBM 3745, 3720, 3725 and 3705 Communications Controllers. 37***mm*** is listed whenever substitution for a particular communication controller is appropriate.
37XXNAS	X.25 Network Access Support for 37XX Communication Controllers. This collection of 37XX software packages allows personal computers, asynchronous TTY terminals, BSC 2780/3780 contention mode terminals, BSC 3270 terminals and native SNA PU type 1and 2 terminal equipment to access host systems via packet switched networks that utilize X.25 link access procedures. From the host's viewpoint, there is no distinction between directly connected terminals and those that communicate via an X.25 network. 3745 products provided by Comm-Pro Associates, Inc.
ABEND	Abnormal End of Task. HNAS issues **NASHALT** or **HALT AT LOC** sysprint messages prior to program termination when unrecoverable errors are detected.
ABM	Asynchronous Balance Mode. Handshake method used to establish an X.25 DTE to DCE connection. Balanced mode allows either the DTE or DCE to initiate the connection as peer entities.
ACB	Application Control Block. VTAM control block generated from the Major Node APPL statement that is used to manage LU sessions.
AMNF	VTAM Application Major Node File. This file contains APPL statements that HNAS uses to provide paths into host applications. Each APPL statement represents an SLU that can be acquired by host applications (PLUs). HNAS opens each APPL for host communication when a path is required for a remote terminal.

APAR	Authorized Program Analysis Report (maintenance). Typically a memo describing a software or operational problem often including a reference to corrective logic (PTF).
ARM	Asynchronous Response Mode. Handshake method used to establish an X.25 DTE to DCE connection. Response mode allows the DCE to control the DTE to DCE connection.
ARP	Address Resolution Protocol. Protocol used to derive an unknown LAN address using a known IP address on an Ethernet or Token Ring network.
BIN (*.BIN)	Binary File Type. Comm-Pro generally refers to EBCDIC (LRECL=80, RECFM=F|FB) sequential files as binary EBCDIC files and assigns the *.BIN designation to filename extension. An example of this is the HNASRCV.BIN distribution file type.
BMT	Bench Mark Test.
BSC	Binary Synchronous Communications line control.
BPAD	BSC Packet Assembler/Disassembler Protocol. Protocol used to transport BSC 2780/3780 RJE work station traffic across X.25 networks.
Callin	Network Initiated Host Access
Callout	Host Initiated Network Access
CART	Command And Response Token Used for Operator message routing.
CC/VRM	Change Control / Version Release Modification Level Identifier. Comm-Pro employs this method of identification for product maintenance and enhancement activity.
CCITT	International Telegraph and Telephone Consultative Committee. Organization sets international standards for data communications (X.21, X.25, X.3 etc.)
CCU	Communications Control Unit. Indicates a 3705, 3720, 3725 or 3745 front end processor.
CDF	Configuration Data File. This file contains definition statements that HNAS uses to create the resource control blocks it needs to manage sessions between X.25 DTEs and host applications.

CID	Connection Identifier An identifier assigned by VTAM for a PLU/SLU session.
CSI	Consolidated Software Inventory is a system used to manage and identify software products typically installed under the control of SMP/E.
CTCP	Communication and Transmission Control Program. Host resident software that utilizes NPSI GATE functions to communicate with devices across an X.25 network.
CUD	Call User Data. The data portion of an X.25 Call Request packet. The CUD field follows the address and facilities fields in the packet.
CUD0	Call User Data, byte 0.
CUME	Cumulative Maintenance. Refers to product distribution levels with cumulative maintenance included up through a specific date.
DASD	Direct Access Storage Device. Storage device for locating programs and data.
DCB	Dataset Control Block. Operating System element that used to manage (DASD) I/O transfers.
DCE	Data Circuit Terminating Equipment. Normally associated with the network component of an X.25 access line.
DLSw	Data Link Switching. A method used in transporting SNA (LLC type 2) data across a router based network.
DNIC	Data Network Identification Code. The first 4-digits of an X.25 DTE address. The DNIC serves to identify the specific public or private packet switched network over which data traffic flows.
DATE	Dedicated Access Transport Extension. Specialized X.25 NPSI extended function that allows host applications to control the X.25 Packet Layer, security control, resource allocation, accounting as well as the X.25 virtual circuit session establishment and disconnect using a control LU session and data transfer using a separate data LU session.
DSLU	Downstream Logical Unit. Protocol used to transport native SNA traffic across TCP/IP networks.

DSP	Display System Protocol. Protocol used to transport BSC 3270 display and printer device traffic across X.25 networks. The PID (protocol id) for DSP is 57.
DTE	Data Terminal Equipment. Normally associated with the router component of an X.25 access line.
ETHERNET	A local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and delayed retransmission. Ethernet uses carrier sense multiple access with collision detection.
FEP	Front End Processor. Normally refers to the family of IBM communications control units (3705, 3720, 3725, 3745). See also CCU/37XX.
FMD	Function Management Data PIU. Component of SNA.
FMID	Function Modification Identifier. A unique name used to identify a function (in the case of HNAS, a program) installed using the IBM SMP/E program product. The identifier assigned to Comm-Pro is LNS. HNAS at the V2R3M0 level is installed with FMID=LNS0230.
GATE	Generalized Access Transport Extension. NPSI function that allows host applications to control X.25 virtual circuit session establishment and disconnect using a control LU session and data transfer using a separate data LU session.
GATEFC	GATE Fast Connect. NPSI function that allows host applications to control X.25 virtual circuit session establishment, disconnect and data transfer using a single LU session. GATEFC provides a faster connection service than standard GATE.
GTF	Generalized Trace Facility. IBM program product used for capturing network trace information.
HDLC	High Level Data Link Control. This protocol is used primarily for X.25 link level communications. This protocol is a superset of SDLC.
HNAS	Host Network Access Support. Comm-Pro’s X.25 Host Network Access Support software. Provides support for Cisco XOT and IBM XTP routers via the host resident TCP/IP stack (NPSI and 37XX communication controller replacement product running in the host). HNAS is a NPSI replacement product.

host-to-host	Host to Host Application Sessions. HNAS provides support for HNAS-XOT to HNAS-XOT host application sessions with no requirement for X.25 facilities or router XOT services.
HOSTNAS	See HNAS.
IP	Internet Protocol.
IPAD	ITI Packet Assembler/Disassembler. Hardware, software or firmware device that provides the interface between non-SNA equipment and an X.25 network. IPAD peers pass non-SNA data between each other in ITI packets. NPSI also has an Integrated PAD function which provides X.3 and X.29 services for ITI devices.
ISARX25	ISARX25 3745 "NPSI like" product in Spain.
ITI	Interactive Terminal Interface. Protocol used to transport ASCII TTY device data across X.25 networks. The PID (protocol id) for ITI is 01 (00 in some environments).
LAN	Local Area Network. A local-area network connects together several machines that are located nearby (in the same room or building) allowing them to share files and devices such as printers.
LAP	Link Access Procedures. Used to identify asynchronous response mode link operation.
LAPB	Link Access Procedures Balanced Mode. Used to identify asynchronous balanced mode link operation.
LCN	Logical Channel Number. Used to identify a virtual circuit multiplexed across an X.25 access link.
LLC	Logical Line Control. Type of framing used to communicate with stations on an Ethernet LAN. LLC is also used to describe the procedures used to support virtual resources. For example, NPSI LLC0 (PCNE) identifies the LLC used to map non-SNA ITI devices to SNA LUT1 resources.
LSA	Link Services Architecture.
LU	Logical Unit (SNA terminal or end user). Indicates an SNA terminal, device or end user.
LUB	Logical Unit Block. The HNAS control block that is used to manage a Logical Unit session.

MCH	Multi-Channel Link. The acronym that represents a NPSI X.25 link resource. HNAS REMOTE TYPE=MCH client.
MXT	Multi-Channel Link Extension. HNAS REMOTE TYPE=MXT client.
NAS	Network Access Support. Comm-Pro's Network Access Support software. Provides enhanced support under NCP and/or EP in the IBM 37XX (3745) communication controller environment.
NCP	Network Control Program. IBM's FEP (37XX) control program for SNA terminal support.
NPSI	Network Packet Switching Interface. IBM's X.25 program product that runs in a FEP as an NCP add on.
NSC	Native Subchannel. The host subchannel that connects a mainframe to a FEP channel adapter.
NVT	Network Virtual Terminal. Normally associated with UNIX systems as a TCP/IP protocol for handling internet TELNET session.
OSA	Open Systems Adapter. A mainframe hardware interface that is used to connect an Ethernet or Token Ring LAN to a host.
PAD	Packet Assembler/Disassembler. Hardware, software or firmware device that provides the interface between a DTE and an X.25 network. See IPAD
PCNE	Protocol Conversion for non-SNA Equipment. NPSI function that allows ITI devices to access VTAM applications as SNA 3767 LUT1 devices.
PIU	Protocol Information Unit. Element used to carry SNA protocol information.
PID	Protocol Identifier. Under X.25 packet level, the P.I.D. is located in the first byte of CUD (call user data) and is used to convey the session connect type.
PP-digits	Port Digits in X.25 Called Address. Old Datapac X.25 expression denoting the least significant digits (normally the last byte) in the called address field in the X.25 Call Request packet. Now generally referred to as the subaddress (SUBD) value.

PPP	Point-to-Point Protocol. Synchronous protocol used to connect internet devices (e.g., personal computers) to an internet provider like NETCOM. Data is moved across the PPP link using TCP/IP. PPP is the successor to SLIP.
PSH	Packet Switched Header. Protocol used to transport native SNA device traffic across X.25 networks. The devices are normally connected via network interface adapters. PSH is the predecessor to QLLC. The PID (protocol id) for PSH is C2.
PTF	Program Temporary Fix (maintenance).
PU	Physical Unit. Indicates an SNA control unit.
PVC	Permanent Virtual Circuit. Logical channel initialized via a Data packet or auto connect timer process.
QLLC	Qualified Logical Link Control. Protocol used to transport native SNA device traffic across X.25 networks. The devices are normally connected via SNA packet assembler/disassembler equipment like the 2216 router. QLLC is the successor to PSH. The PID (protocol id) for QLLC is C3.
QPAD	QLLC Packet Assembler/Disassembler. Hardware, software or firmware device that provides the interface between SNA equipment and the X.25 network. QPAD peers pass SNA data between each other in QLLC packets.
RARP	Reverse Address Resolution Protocol. Protocol used to derive an unknown IP address using a known LAN address on an Ethernet or Token Ring network.
README	A file or member containing special instructions, additional information or supplemental documentation.
RH	Request/Response Header. The RU header portion of a PIU.
RU	Request/Response Unit. The command and/or data portion of a PIU.
SDLC	Synchronous Data Link Control. This protocol is used primarily for IBM SNA link level communications.
SLIP	Synchronous Line Interface Protocol. Synchronous protocol used to connect internet devices (e.g., personal computers) to an internet provider like USINET. Data is moved across the SLIP link using TCP/IP. SLIP is the predecessor to PPP.

SMP/E	System Modification Program Extended. An IBM program used to install and maintain operating system components. HNAS 230 and above may be installed using SMP/E
SNA	System Network Architecture. The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.
SPU	Secondary Physical Unit. HNAS REMOTE TYPE=SPU client specifically for QLLC resources.
Staging Datasets	Staging Datasets. Staging datasets are sequential files that are used during the intermediate steps of the HNAS installation process (see HNASRCV JOB). The staging datasets are used in the process of preparing the HNAS distribution libraries which are in partitioned dataset format.
STR (*.STR)	Stream File Type. Comm-Pro refers to binary distribution file types as stream (STR) files and assigns the *.STR designation to the filename extension. In HNAS 211 and earlier, the edistribution and CDROM distribution files were binary EBCDIC (LRECL=80,RECFM=F|FB) sequential files containing the HNAS object, macro, control or JCL members. In the initial release of HNAS 220 (prior to 12-02-2002) the distribution files were either binary EBCDIC (LRECL=80,RECFM=F|FB) or TSO XMIT generated unloaded partitioned dataset files. In the current HNAS 220 release, stream files now only refer to binary TSO XMIT generated unloaded partitioned dataset files. The *.STR file types are used in our edistribution, CDROM product distribution or product maintenance.
SUBD	Subaddress Digits. The last byte of the called DTE address in an X.25 Call Request packet. The SUBD byte is in packed decimal format.
SVC	Switched Virtual Circuit. Logical channel initialized via Call Request packet.
symbol	Assembler language symbol. A valid assembler language symbol is a string of from 1 to 8 alphanumeric characters. The first character only must be alphabetic and not numeric. Alphabetic characters are the letters A through Z and $, #, @. Alphanumeric characters are all of the alphabetic characters plus numerics 0 through 9.
SYSCONS	System Console. The Operating System (z/OS, OS/390, MVS) master console.

SYSGEN	System generation process. An older term referring to the process of generating the environment for a mainframe system or 37xx load module.
SYSPRINT	System Print Log. The primary Operating System (z/OS, OS/390, MVS) SYSOUT dataset.
TCB	Task Control Block. The operating system element that controls the amount of CPU processor time that is given to a program or process.
TCD	Used in reference as to the date that a product distribution tape was generated or created.
TCP	Transmission Control Protocol. One of a number of second level IP protocols that is used to transport end user data across an IP network.
TOKEN RING	A network that uses ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission. Each machine can transmit only while it is holding the token.
TXT (*.TXT)	Text File Type. Comm-Pro generally refers to ASCII (CR or CRLF delimited LRECL=80, RECFM=F or RECFM=V) files as ASCII text files and assigns the *.TXT designation to filename extension. An example of this is the HNAS-RCV.TXT distribution file which is suitable for viewing on a PC.
VC	Virtual Circuit.
VCB	Virtual Circuit Block. The HNAS control block that is used to manage a Virtual Circuit session.
VCN	Virtual Circuit Number. An identifier assigned by the X.25 network for a Virtual Circuit. See also LCN.
WAN	Wide Area Network. A wide-area network is a set of widely separated computers connected together.
WTO	Write to operator. Host macro used to display data on the mainframe’s operator console.
WTOR	Write to operator with reply. Host macro used to display data on the mainframe’s operator console and request operator input.

X.25	Interface between DTE and DCE for terminals operating in the packet mode on public and private data networks.
X.28	DTE/DCE interface for start-stop mode DTE accessing the packet assembly/disassembly facility in a public or private network situated in the same country.
X.29	Procedures for the exchange of control information and user data between a packet assembly disassembly facility and a packet mode DTE or another PAD.
X.3	Packet assembly/disassembly facility in a public or private data network.
X.75	Terminal and transit call control procedures and the data transfer system on international circuits between packet switched data networks.
XOT	X.25 over TCP/IP. Open Cisco protocol used to transfer X.25 data across a TCP/IP router based network. HNAS REMOTE TYPE=XOT client.
XPAD	Transparent Packet Assembler/Disassembler. NPSI function that allows host applications to control ITI PAD parameters.
XTP	X.25 through TCP/IP. Proprietary IBM protocol used to transfer X.25 data across a TCP/IP router based network. HNAS REMOTE TYPE=XTP client.
ZAP	Runtime module patch.
ZIP (*.ZIP)	File type designation for a compressed archive file. Content can be ASCII, EBCDIC or Binary.

HNAS V2R4M0 Index

Symbols

Numerics

C

E

F

G

M

COMM-PRO ASSOCIATES
HOST NETWORK ACCESS SUPPORT
INSTALLATION AND CONFIGURATION

READER'S COMMENT FORM

This manual is the primary document for the installation and configuration of Comm-Pro Associates' Host Network Access Support Program Product. It was designed to serve as a reference source for systems programmers who are responsible for the creation and maintenance of the Host Network Access Support software.

This reader comment form may be used to communicate your opinions about this document. They will be reviewed and, when applicable, will cause changes to be incorporated into the document.

Comm-Pro may use and distribute any of the information that you supply in any way it believes appropriate without incurring any obligation whatsoever.

Possible topics for comment are: clarity, accuracy, completeness, organization and legibility.

If you do not wish to comment specifically on any of the topics listed above, we would appreciate your response to the following questions:

Based on your experience, please rate this document:

very good | fair | poor | very poor

What is your job title?

____

If you choose to reply, please give your name, mailing address and date:

____

____

____

____

COMM-PRO ASSOCIATES
HOST NETWORK ACCESS SUPPORT
INSTALLATION AND CONFIGURATION

READER'S COMMENT FORM

fold and tape fold and tape fold and tape

Comm-Pro Associates Inc.
Attn: Reader's Comment
25852 McBean Parkway #611
Santa Clarita, California 91355

fold and tape fold and tape fold and tape