Comm-Pro Associates
Host Network Access Support
V2R4M0

Chapter 2

Installation and Activation Procedures

This file revised June 30, 2011 5:19 pm, includes maintenance thru 2400110.

Contact Information

Phone:	(661) 284-3650
Fax:	(661) 287-1646
Email:	support@comm-pro.com sales@comm-pro.com
FTP:	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.

Chapter 2 Table of Contents

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 HNAS-RCV 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***.HNAS-LOAD. 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. ***hlq****1* is the first qualifier in ***hlq***. If ***hlq*** has only one level then ***hlq****1* 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***.SLNSCNTL(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 communi-

cation 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#,''YYYYYYYYYYYYYYY'',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 ln*sxxxx*.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.

We strongly recommend that you use a test or development environment to perform the initial HNAS testing independently of the production environment. 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.

This condition does not affect the majority of HNAS callout users 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.

Router Environment Considerations (Cisco)

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 VTAM and TCPIP stack interrupts. These exits pass information from the VTAM and TCPIP exits to the HNAS taks level for processing.

The VTMEXIT module processes VTAM interrupts.

The NASTCP module provides a similar function for TCPIP stack interrupts.

Since these modules are VTAM and TCPIP level dependent, they are assembled as part of the HNAS installation procedure to ensure they match the level of VTAM and TCPIP you are running.

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 com-

municating 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,...,*sp*n)	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 {ALLON|ALLOFF}
{MAXDATA|MINDATA|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.

ALLON (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.

MINDATA (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 {ALLON|ALLOFF}
{ICR|ICRF|OCR|ICLR|OCLR}

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

ALLON (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.

ICRF (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 {ALLON|ALLOFF}

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

ALLON (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 {ALLON|ALLOFF}
{MAXDATA|MINDATA]|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.

ALLON (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).

ALLON (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**
TRCCNFG	**TCFG**	**TRCCONS**	**TCON**

Table 2: Abbreviated Start Parameters

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

Equivalent or Similar Console Command	DPARM EXEC Display Output
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.	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}
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	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}
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.	TRCBFR {ALLON|ALLOFF} TRCBFR {CONS|NETV|TCP| TMR|UTIL|XOT|XTP|NOTYPES}
TRCBFRQ {ON|OFF}	TRCBFRQ {ON|OFF}
TRCBST {ON|OFF}	TRCBST {ON|OFF}
TRCCONS {ON|OFF}	TRCCONS {ON|OFF}
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.	TRCDATA {ALLON|ALLOFF} TRCDATA {CONS|NETV|TCP| TMR|UTIL|XOT|XTP|NOTYPES}

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..
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// 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..
12345678901234567890123456789012345678901234567890123456789012345678901234567890
// 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:

NAS3799I Without Compression (SHOWCMSG OFF):

```
0         1         2         3         4         5         6
01234567890123456789012345678901234567890123456789012345678901234567890123456789
```

```
|                                                                      |
NAS3799I LU MCH1PVC  ENDING   SESSION ON MCH MCH1     RMT  CAUSE/DIAG=
000/130 (00/82) DIAGX=0000
|                          |
012345678901234567890123 45
7         8         9
```

NAS3799I With Compression (SHOWCMSG ON):

```
0         1         2         3         4         5         6
01234567890123456789012345678901234567890123456789012345678901234567890123456789
|                                                                              |
NAS3799I LU MCH1PVC ENDING SESSION ON MCH MCH1 RMT CAUSE/DIAG=000/130
(00/82) DIAGX=0000
|                 |
012345678901234567
7         8
```

Note: While message compression does use additional CPU cycles, the affect appears to be minimal. However, HNAS environments with excessive alert message activity should consider measuring the effect upon CPU load with compression enabled (SHOWCMSG ON) and disabled (SHOWCMSG OFF) to see if compression is causing an unacceptable percentage of CPU load. Our in-house testing revealed a CPU seconds difference per WTO for SHOWCMSG ON versus SHOWCMSG OFF to be an additional 0.000011048 seconds per WTO or approximately 0.002846% additional overhead per WTO.

Note: The SHOWCMSG support described above was introduced into 240 with APAR 2400036.

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.

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.commpro.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 SVC Call Request and PVC Setup 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 SVC Clear Request and PVC Setup Rejection 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=	0 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=	 60 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= 0 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.	

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 re-issue 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 2 Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

S

T

U

V

W

X

Z