



H210IPZ NetEx/IP[®] and TNP210
for IBM z/OS Operating Systems

Release 7.4

Installation Reference Manual

Revision Record

Revision	Description
01 (Sep. 2000)	<ul style="list-style-type: none"> Initial manual release by Network Executive Software Manual updated to correspond to release 5.0. Adds information for support of NetEx over IP networks.
02 (Apr 2001)	<ul style="list-style-type: none"> Manual updated to indicate Parallel channels are also supported on NESiGate device.
03 (Aug 2001)	<ul style="list-style-type: none"> Miscellaneous editorial changes. Manual prepared for electronic distribution.
04 (Jan 2002)	<ul style="list-style-type: none"> Manual updated to correspond to NetEx/IP release 6.0.
05 (Apr 2002)	<ul style="list-style-type: none"> Manual updated to correspond to NetEx/IP release 6.1.
06 (Jan 2009)	<ul style="list-style-type: none"> Manual updated to correspond to H210IPZ NetEx/IP release 7.0. Installation Reference Manual split from Software Reference Manual
07 (Feb 2010)	<ul style="list-style-type: none"> Manual updated to correspond to H210IPZ NetEx/IP release 7.0.5
7.1 (Feb 2011)	<ul style="list-style-type: none"> Manual updated to correspond to H210IPZ NetEx/IP release 7.1
7.1-01 (May 2011)	<ul style="list-style-type: none"> Correction of the accepted value range for the “MAXODATA” initialization parameter. Manual updated to include description of sharing license key file Correction to the description of the “HOST NAME” initialization parameter.
7.2 (Mar 2012)	<ul style="list-style-type: none"> Updated descriptions and examples of TNP configuration Added TNP startup options so it can be started as either a batch job or a Started Task Added more detailed description to SET GTRACE initialization statement. Updated to correspond to H210IPZ NetEx/IP release 7.2
7.2-01 (Feb 2013)	<ul style="list-style-type: none"> Updated to correspond to H210IPZ NetEx/IP release 7.2.1
7.2-02 (Dec 2013)	<ul style="list-style-type: none"> Fixed minor typographical errors
7.3 (Sept 2014)	<ul style="list-style-type: none"> Updated to correspond to H210IPZ NetEx/IP release 7.3
7.3-01 (Sept 2014)	<ul style="list-style-type: none"> Added documentation for NTXDLOAD installation library
7.4 (Mar 2015)	<ul style="list-style-type: none"> Updated to correspond to H210IP NetEx/IP release 7.4

Minor editorial revisions are not indicated.

© 2002-2015 by Network Executive Software. Reproduction is prohibited without prior permission of Network Executive Software. Printed in the U.S.A. All rights reserved.

Preface

This manual describes how to use and install the H210IPZ NETwork EXecutive (NetEx®) software for the IBM z/OS operating systems. It also describes the general NetEx® program architecture, session concepts, and Application Program Interfaces (APIs). Readers are not expected to be familiar with NetEx before using this manual. However, writing programs to the NetEx APIs requires an understanding of programming in the supported languages; usage of NetEx requires an operational understanding of the host operating system.

The *H210IP Release 6.1 Reference Manual* was separated into the following manuals for H210IPZ:

H210IPZ NetEx/IP for IBM z/OS Operating Systems Installation Reference Manual

H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual

H210IPZ NetEx/IP for IBM z/OS Operating Systems Programming Reference Manual

The following manual had been previously separated out and remains that way for H210IPZ:

H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages & Abend Codes Reference Manual

Brief description of product reference manuals

A brief overview of the various reference manuals follows.

- The *Operator Reference Manual* is intended for those responsible for daily operations of H210IPZ on z/OS.
- The *Programming Reference Manual* is intended for those responsible for using the NetEx/IP API's.
- The *Systems Messages & Abend Codes Reference Manual* documents the error messages, NRBSTAT error codes, and Abend codes that may be issued by H210IPZ.
- The *Installation Reference Manual* is intended for those responsible for installing, configuring, and maintaining H210IPZ on z/OS, and contains the following information:
- “H210IPZ NetEx/IP Overview” describes NetEx/IP and is intended for all readers.
- “NetEx/IP IP Addressing” describes NetEx/IP IP addressing concepts.
- “NetEx/IP Pre-Installation” contains pre-installation information and requirements pertaining to the various configurations supported by H210IPZ.
- “Installing NetEx/IP” describes how to install H210IPZ NetEx/IP.
- “Configuration Manager” describes how to run the Configuration Manager.
- “Initialization Statements” describes how to define initialization parameters to NetEx/IP.
- “User Exits” describes how to use user exits to maintain additional control over the connection of NetEx applications and the movement of data between them.
- “Appendix A. Glossary” lists and describes commonly used terms and acronyms in this document.

Reference Material

Reference material may be found in the following publications:

Number	Title and Description
MAN-M&A-H210IPZ	H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages and Abend Codes
MAN-OPR-H210IPZ	H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual
MAN-API-H210IPZ	H210IPZ NetEx/IP for IBM z/OS Operating Systems Programming Reference Manual
460195	H211 Bulk File Transfer (BFX™) Utility for IBM MVS Software Reference Manual
460201	H212R Print File Transfer (PFX™) Receiver Utility for IBM MVS Systems Installation and User Guide
460345	H212T Print File Transfer (PFX) Utility for IBM MVS Systems Software Reference Manual
MAN-REF-EFT213	eFT for z/OS Operating System User Guide
MAN-CNET-CONFIG-MGR	“C” Configuration Manager and NetEx Alternate Path Retry (APR) User Guide

Notice to the Reader

The material contained in this publication is for informational purposes only and is subject to change without notice. Network Executive Software is not responsible for the use of any product options or features not described in this publication. It assumes no responsibility for any errors that may appear in this publication. Refer to the revision record (at the beginning of this document) to determine the revision level of this publication.

Network Executive Software does not, by publication of the descriptions and technical documentation contained herein, grant a license to make, have made, use, sell, sublicense, or lease any equipment or programs designed or constructed in accordance with this information.

This document may contain references to the trademarks of the following corporations:

Corporation

Network Executive Software

International Business Machines Corp.

Trademarks and Products

NetEx®, BFX™, PFX™, USER-Access™

IBM, z/OS, VTAM, ESCON

These references are made for informational purposes only.

The diagnostic tools and programs described in this manual are not part of the products described.

Notice to the Customer

Comments may be submitted over the Internet by addressing email to:

support@netex.com

or, by visiting our web site at:

<http://www.netex.com>

Always include the complete title of the document with your comments.

Document Conventions

The following notational conventions are used in this document.

Table 1. Documentation Conventions	
Format	Description
displayed information	Information displayed on a CRT (or printed) is shown in this font .
<i>user entry</i>	<i>This font</i> is used to indicate the information to be entered by the user.
UPPERCASE	The exact form of a keyword that is not case-sensitive or is issued in uppercase.
MIXedcase	The exact form of a keyword that is not case-sensitive or is issued in uppercase, with the minimum spelling shown in uppercase.
bold	The exact form of a keyword that is case-sensitive and all or part of it must be issued in lowercase.
lowercase	A user-supplied name or string.
<u>value</u>	Underlined parameters or options are defaults.
<label>	The label of a key appearing on a keyboard. If "label" is in uppercase, it matches the label on the key (for example: <ENTER>). If "label" is in lowercase, it describes the label on the key (for example: <up-arrow>).
<key1><key2>	Two keys to be pressed simultaneously.
No delimiter	Required keyword/parameter.

Contents

Revision Record	i
Preface.....	iii
Brief description of product reference manuals	iii
Reference Material	iv
Notice to the Reader.....	iv
Notice to the Customer	iv
Document Conventions.....	vi
Contents	vii
Figures.....	xviii
Tables	xx
H210IPZ NetEx/IP Overview.....	1
General.....	1
New Features	2
H210IPZ Release 7.4	2
REUSASID Support	2
USERPK/USERSK Initialization Statements.....	2
Other Features and Miscellaneous Additions and Corrections	2
H210IPZ Release 7.2.....	2
TNP	2
Miscellaneous corrections.....	2
Software Key	2
H210IPZ Release 7.1	3
TNP	3
Additional OSA IP Configuration Option	3
New Software Key Format	3
Retry During NetEx/IP Initialization	3
Installation Procedure Change	3
H210IPZ Release 7.0.....	3
H210IP Release 6.1.....	4
NetEx Characteristics.....	5
External Interface.....	5
Internal Interaction.....	5
NetEx Connections	5
Design Efficiency and Flexibility	5
Block Segmenting.....	6
Alternate Path Retry.....	6
Remote Operator Interface.....	6
User Exits.....	6
Basic I/O Flow	6
NetEx and the ISO Model.....	7
Session Layer Services	9
Transport Layer Services	9
Network Layer Services.....	10

Driver Sublayer Services.....	10
NetEx/IP IP Addressing.....	11
Overview	11
OSA	11
NetEx Network Address.....	11
NetEx IP-hostname.....	11
IP Name Resolution.....	11
Virtual IP Address (VIPA) Support	12
OSA NCT Requirement	12
OSA NTXINIT Initialization File Requirement.....	12
NetEx/IP Pre-Installation	13
General	13
Prerequisites	14
z/OS Considerations	15
ALLOWUSERKEYCSA Support.....	15
REUSASID Support.....	15
SMF Interface.....	15
SMF Installation	16
Example.....	16
Related Topics	16
SMF Record Layout	17
OSA Configuration Requirements.....	18
CNTLUNIT Statement	18
IODEVICE Statement	18
Device I/O Definition Requirements.....	18
OSA-Express Device.....	18
OSA Device Definitions.....	18
VTAM Considerations	19
TCP/IP Considerations	19
RACF Considerations.....	20
Policy Agent Considerations	20
OSA NetEx/IP Configuration Examples	20
Examples of NetEx/IP Non-VIPA configurations	21
Example 1: One physical ethernet interface.....	21
Example 2: Two physical ethernet interfaces.....	23
Example of VIPA configuration.....	26
Dynamic VIPA configuration	30
Alternate Path Retry	31
Alternate Path Retry Programming Considerations	31
TNP.....	32
TNP Configuration	33
H210IPZ Configuration.....	33
tnpop.....	36
Requester Configuration.....	38
Installing NetEx/IP	39
General	39
Related Topics.....	39
Release Distribution	40

Distribution Contents	40
NTX0704	40
NTXINST	40
LPA Module Considerations	40
Before Installing H210IPZ.....	40
Related Topics	41
Obtaining the Software Key.....	42
Installation Process	44
Step 1. Obtain the H210IPZ distribution file.....	44
Step 2. Upload the distribution file to z/OS.....	44
Step 3. TSO RECEIVE the distribution file.	45
Step 4. Execute the NTXINST job on z/OS.	45
Step 5. Check for required updates.....	57
Step 6. Obtain the H210IPZ software key.	57
Step 7. Review the H210IPZ initialization parameters.....	58
Step 8. Review the NCT configuration.....	58
Step 9. Authorize the NTXLOAD load library.....	59
Step 10. Activate the NetEx/IP LPA modules.....	59
Step 11. Define OSA devices to VTAM.....	60
Step 12. Define Device and Link statements to TCP/IP	60
Step 13. Define RACF OMVS segment for Netex	60
Step 14. (Optional) Define RACF OMVS segment for TNP	60
Step 15. (Optional) Update Policy Agent	60
Step 16. Update DNS.....	60
Step 17. Activate VTAM Nodes.....	60
Step 18. (Optional) Add NETEX entry in Program Properties Table.....	61
Step 19. Review Installed JCL.....	61
Step 20. Start NETEX.....	61
Step 21. (Optional) Execute the NTXMEAT Program.....	62
Step 22. (Optional) Execute the NTXMGGEN Program	62
Subsystem Validity Checking	63
Emergency Override Procedure	63
Configuration Manager.....	65
General.....	65
OSA NCT Considerations	65
Sample NCT Definitions	65
Configuration Manager Control Statements	67
DESELECT	69
Related Topics	69
EXIT	69
MAKEPAM.....	70
Related Topics	70
SELECT.....	71
Related Topics	71
Initialization Statements.....	73
General.....	73
Initialization Statements.....	73
ALIAS.....	77
Example	77

Related Topics	77
BACKOFF	78
Example	78
BITMINT	78
Example	78
CFOTM	79
Example	79
Related Topics	79
CMDNO	79
Example	79
CMDPRE	80
Example	80
CONFIG	81
Example	81
Related Topics	81
CONTIME	82
Example	82
Related Topics	82
DDMAXMLT	82
Example	82
Related Topics	82
DEADTIME	83
Example	83
Related Topics	83
DEFBI	84
Example	84
Related Topics	84
DEFBO	84
Example	84
Related Topics	84
DESCCD	85
Example	85
Related Topics	85
DFRT	85
Example	85
Related Topics	85
DIRE	86
Example	86
Related Topics	86
DISE	86
Example	86
Related Topics	86
DOWD	87
Example	87
Related Topics	87
DQHB	88
Example	88
Related Topics	88
DQLB	89
Example	89
Related Topics	89

DRIVAVG	89
Example	89
Related Topics	89
DRIVLIM	90
Example	90
Related Topics	90
DRIVMAX	91
Example	91
Related Topics	91
ENDINI1	91
Example	91
FMSML	92
Related Topics	92
FUBS	92
Related Topics	92
GTRACE	92
Related Topics	93
HOST NAME	93
Example	93
Related Topics	93
IDLEMAX	93
Example	94
Related Topics	94
IDLETIME	94
Example	94
Related Topics	94
INACTIVE	94
Example	95
Related Topics	95
IOCOUNT	96
Related Topics	96
LARNAME	96
Example	96
LOGCMD	97
Example	97
Related Topics	97
MAXBI	97
Example	97
Related Topics	97
MAXBO	98
Example	98
Related Topics	98
MAXDDBQ	98
Example	98
Related Topics	98
MAXDREAD	98
Example	99
Related Topics	99
MAXDWRT	99
Example	99
Related Topics	99

MAXODATA.....	100
Example.....	100
Related Topics.....	100
MAXSEG.....	101
Example.....	101
Related Topics.....	101
MBFI.....	102
Example.....	102
Related Topics.....	102
MBFO.....	102
Example.....	103
Related Topics.....	103
MFCP.....	103
Example.....	103
Related Topics.....	103
MIHTM.....	104
Example.....	104
Related Topics.....	104
MKBS.....	104
Example.....	104
Related Topics.....	104
MSGVL.....	105
Example.....	105
Related Topics.....	105
MSGML.....	106
Example.....	106
Related Topics.....	106
MSGTM.....	107
Example.....	107
Related Topics.....	107
MSGXL.....	108
Example.....	108
Related Topics.....	108
MULTIH.....	109
Example.....	109
Related Topics.....	109
NTXOPER.....	110
Example.....	110
Related Topics.....	110
OSAA.....	111
Related Topics.....	111
OSA1IF.....	111
Related Topics.....	111
OUTCLASS.....	112
Related Topics.....	112
PAGENIT.....	112
Example.....	112
Related Topics.....	112
PAGESAVE.....	113
Example.....	113
Related Topics.....	113

PAGEXTRA	114
Example	114
Related Topics	114
PAGExxx	115
Example	116
Related Topics	116
PDQNAME	116
Example	116
PRFP	117
Example	117
Related Topics	117
RDDP	118
Example	118
Related Topics	118
RDII	118
Example	118
Related Topics	118
READTIME	119
Example	119
Related Topics	119
ROUTCD	120
Example	120
Related Topics	120
RQPH	120
Example	120
Related Topics	120
RQPL	121
Example	121
Related Topics	121
RQPS	122
Example	122
Related Topics	122
RRCI	123
Example	123
Related Topics	123
SESAVG	123
Example	123
Related Topics	123
SESLIM	124
Example	124
Related Topics	124
SESMAX	124
Example	124
Related Topics	124
SMFTYPE	125
Example	125
Related Topics	125
SMWTM	125
Example	125
Related Topics	125
SNDGRNM	126

Example.....	126
SRBNAME.....	126
Example.....	126
SRCI.....	127
Example.....	127
Related Topics.....	127
SRPH.....	127
Example.....	127
Related Topics.....	127
SRPL.....	128
Example.....	128
Related Topics.....	128
SRPS.....	128
Example.....	128
Related Topics.....	128
SRUI.....	129
Example.....	129
Related Topics.....	129
SSINAME.....	129
Example.....	129
STRP.....	130
Example.....	130
Related Topics.....	130
SUBSYS.....	130
Example.....	130
SWTNAME.....	131
Example.....	131
TRACE.....	132
Examples.....	132
Related Topics.....	134
TRCNO.....	134
Example.....	134
Related Topics.....	134
TRCSIZE.....	135
Example.....	135
Related Topics.....	135
TRMNAME.....	135
Example.....	135
TWOPART.....	136
Example.....	136
Related Topics.....	136
UDPP.....	136
Related Topics.....	136
UIFNAME.....	137
Example.....	137
USERNAME.....	137
Example.....	137
USERPK.....	138
Example.....	138
USERSK.....	138
Example.....	138

WDOGIN	139
Example	139
Related Topics	139
XCONF	140
Example	140
Related Topics	140
XCONN	141
Example	141
Related Topics	141
XCXN	142
Example	142
Related Topics	142
XDATAI	143
Example	143
Related Topics	143
XDATAO	144
Example	144
Related Topics	144
XDISC	145
Example	145
Related Topics	145
XINIT	146
Example	146
Related Topics	146
XOFFR	147
Example	147
Related Topics	147
XTERM	148
Example	148
Related Topics	148
User Exits	149
General	149
General Exit Interface	150
Specific Exits	151
NXMUX1IN	152
Entry Specification	152
Entry Conditions	152
Exit Specification	152
Exit Conditions	152
NXMUX1TM	153
Entry Specification	153
Entry Conditions	153
Exit Specification	153
Exit Conditions	153
Restrictions	153
NXMUX2OF	154
Entry Specification	154
Entry Conditions	154
Exit Specification	155
Exit Conditions	155

NXMUX2CN	156
Entry Specification	156
Entry Conditions.....	156
Exit Specification	157
Exit Conditions.....	157
NXMUX2CF	159
Entry Specification	159
Entry Conditions.....	159
Exit Specification	160
Exit Conditions.....	160
NXMUX3CX	161
Entry Specification	161
Entry Conditions.....	161
Exit Specification	163
Exit Conditions.....	163
NXMUX4DI.....	164
Entry Specification	164
Entry Conditions.....	164
Exit Specification	166
Exit Conditions.....	166
NXMUX4DO	167
Entry Specification	167
Entry Conditions.....	167
Exit Specification	168
Exit Conditions.....	168
NXMUX5DC	170
Entry Specification	170
Entry Conditions.....	170
Exit Specification	171
Exit Conditions.....	172
Exit Installation	173
Programming Considerations	173
Standards and Conventions	173
NTCOPT Macro	173
NTCMOD Macro	174
NTCCALL Macro	175
NTCENTER Macro.....	176
NTCLEAVE Macro.....	177
Copy Members	178
Available Services.....	178
Getting Memory	179
Freeing Memory	179
Requesting Time of Day Services	179
Requesting Timer Services.....	180
Miscellaneous Considerations	180
Examples	180
Initialization/Termination Exit.....	181
Offer/Connect Request Exit	183
Connection Exit.....	186

Appendix A. Glossary	191
-----------------------------------	------------

Index.....199

Figures

Figure 1. Basic I/O Flow	7
Figure 2. ISO Model Communication	8
Figure 3. NetEx and the ISO Model	9
Figure 4. ALLOWUSERKEYCSA SCHEDxx member	15
Figure 5. NetEx/IP OSA-Express device definition requirements	19
Figure 6. Sample TRLE Definition	19
Figure 7. Sample TCP/IP OSA-Express Definition	20
Figure 8. Sample TRLE entry with 1 OSA	21
Figure 9. Sample TCP/IP Profile with 1 OSA	22
Figure 10. Sample NetEx/IP NCT with 1 OSA	22
Figure 11. Sample NetEx/IP NTXINIT with 1 OSA and OSA1IF ON	22
Figure 12. Sample NetEx/IP NTXINIT with 1 OSA and OSA1IF off	23
Figure 13. Sample NetEx/IP NTXINIT with 1 OSA and OSAA defines IP	23
Figure 14. Sample TRLE with 2 OSAs	24
Figure 15. Sample TCP/IP Profile with 2 OSAs	24
Figure 16. Sample NetEx/IP NCT with 2 OSAs	25
Figure 17. Sample NetEx/IP NTXINIT with 2 OSAs and OSA1IF ON – Invalid configuration	25
Figure 18. Sample NetEx/IP NTXINIT with 2 OSAs and OSA1IF OFF	26
Figure 19. Sample NetEx/IP NTXINIT with 2 OSAs and OSAA defines IP	26
Figure 20. Sample TRLE with 2 OSAs and VIPA	27
Figure 21. Sample TCP/IP Profile with 2 OSAs and VIPA	28
Figure 22. Sample NetEx/IP NCT with 1 VIPA OSA	28
Figure 23. Sample NetEx/IP NTXINIT with 1 VIPA OSA and OSA1IF ON	29
Figure 24. Sample NetEx/IP NTXINIT with 1 OSA and OSA1IF OFF	29
Figure 25. Sample NetEx/IP NTXINIT with 1 OSA and OSAA defines IP	29
Figure 26. Output display of an OFFERED TNP job	32
Figure 27. Output display of connected TNP jobs	33
Figure 28. Distributed TNP job used when running TNP as a batch job	34
Figure 29. Distributed TNP job used to start a TNP started task (TNPSTART(STCSTART) specified)	34
Figure 30. Distributed TNP job used to start a TNP started task (TNPSTART(STCVS) specified)	35
Figure 31. Distributed TNP Proc	35
Figure 32. Requester host configuration file	36

Figure 33. Sample NCT with NetEx/IP Requester host.....	37
Figure 34. Sample NetEx/IP Requester configuration.....	38
Figure 35. Output display of 'D M=CPU' command.....	42
Figure 36. Sample PRODCONF records	58
Figure 37. Sample LICCODES record	58
Figure 38. Sample NTXMEAT Job, Member 'NTXMEAT' in hlq.NTXCTL.....	62
Figure 39. Sample NTXMGEN Job, Member 'NTXMGEN' in hlq.NTXCTL.....	63
Figure 40. NCT definition for Linux NetEx/IP host.....	66
Figure 41. NCT definition for z/OS NetEx/IP OSA	66
Figure 42. OSAA definition in NTXINIT initialization file	66
Figure 43. Sample Configuration Manager Job Member "CCONFIG" in hlq.NTXCTL.....	68
Figure 44. SMF Record Generation Exit (NXMUX5DC).....	170
Figure 45. NTCMOD for a z/OS System.....	175
Figure 46. NTCENTER Macro for a z/OS System.....	177
Figure 47. Initialization/Termination Exit	183
Figure 48. Offer/Connect Request	186
Figure 49. Connection Exit	189

Tables

Table 1. Documentation Conventionsvi

Table 2. ISO Model 7

Table 3. SMF Record Layout 17

Table 4. Initialization Statements 73

Table 5. Trace Events 133

Table 6. Classes of Trace Events..... 134

Table 7. Parameter List Fields..... 151

Table 8. User Exit Module Names and Entry Points..... 151

H210IPZ NetEx/IP Overview

General

The NETwork Executive (NetEx®) software allows two or more application programs (which may be on different host computers) to communicate with each other at multi-megabit speeds. The NetEx family of software consists of different versions of NetEx for use with different operating systems. All of these versions provide a common Application Programming Interface (API) to simplify programming and portability. Application programs are also available for use with NetEx, such as Bulk File Transfer (BFX™) which provides the capability to move bulk sequential file data from one computer system to another; Print File Transfer (PFX™) which automatically monitors and transfers print files from one system's print spool (queue) to another; and USER-Access®, which provides a powerful remote access, file transfer and scripting capability.

NetEx software resides as a subsystem within each IBM host involved in the communication. NetEx allows communication to take place at any time during host operations, independent of other functions in the system.

Later sections describe the characteristics of NetEx and how it uses the International Standards Organization (ISO) guidelines for open systems interconnection.

New Features

H210IPZ Release 7.4 REUSASID Support

Release 7.2 Patch 5215 added support for the toleration of reusable address spaces. Release 7.4 adds the support to NetEx/IP so it can optionally be run in a reusable address space if so desired. To run NetEx/IP in a reusable address space, ensure the current DIAGxx PARMLIB member specifies REUSASID(YES), which became the system default in z/OS V1R12. NetEx/IP must then be started by adding the REUSASID=YES parameter on the z/OS START command as follows:

```
S NETEX,REUSASID=YES
```

NetEx/IP can still be run in an ordinary address space (i.e. a non-reusable address space) if so desired, by starting NetEx/IP without the REUSASID=YES parameter as follows:

```
S NETEX
```

As a general guideline, it is better to run NetEx/IP in a reusable address space, in order to avoid any potential long-term issue with running out of ordinary address spaces.

USERPK/USERSK Initialization Statements

With the implementation of REUSASID support, it is now necessary to identify Netex user programs that execute in a program key other than key8, or use storage for Netex data buffers that is in a storage key different from the program key. These statements are required whether or not NetEx/IP is running in a reusable address space, but only if the Netex user programs are using program and/or storage keys other than key8.

Note: Since most user programs run in program key 8, these initialization statements are not normally needed.

Other Features and Miscellaneous Additions and Corrections

There are miscellaneous corrections contained in this distribution. Refer to the H210IPZ Release 7.4 Memo-to-Users for a complete description.

H210IPZ Release 7.2

TNP

The TNP component can be deployed as either a batch job or a started task, as specified during the installation process.

Miscellaneous corrections

There are miscellaneous corrections contained in this distribution. Refer to the H210IPZ Release 7.2 Memo-to-Users for a complete description.

Software Key

A new software key is required for this release when upgrading from a Release 7.0 (or prior) installation. When upgrading from a Release 7.1 installation, the same software key will work with Release 7.2.

H210IPZ Release 7.1

TNP

The TNP feature serves as a NetEx/IP “proxy”, which can be used by other hosts for which a host-based NetEx/IP is not available. A NetEx/IP Requester, residing on an HP NonStop, OpenVMS, Bull, or Stratus server, works with BFX, USER-Access, eFT, PFX (or other NetEx/IP applications running on the Requester server), and reads and writes the application’s NetEx/IP requests over a TCP/IP connection to TNP, which then passes the request to H210IPZ, on behalf of the Requester’s NetEx/IP application. In effect, TNP serves as a “proxy” application for the Requester’s NetEx/IP applications.

Prior to H210IPZ with TNP, Requester applications would use a NESiGate-LO device to provide NetEx/IP services. H210IPZ with TNP eliminates the need for NESiGate-LO.

Note: the TNP feature requires OSA as the only network connectivity option. TNP is not supported in NESiGate channel configurations.

Additional OSA IP Configuration Option

An additional configuration option is now available for specifying the local IP address that NetEx/IP should bind to during initialization. A unique IP address can be specified on each of the OSAA initialization statement(s). This may be helpful in Dynamic VIPA configurations, and/or where DNS is not used for NetEx/IP hostname resolution.

New Software Key Format

A new software key is required for this release.

Retry During NetEx/IP Initialization

If NetEx/IP is started before TCP/IP has completed initialization, NetEx/IP will now wait and retry for a period of up to two minutes before giving up.

Installation Procedure Change

SMP/E is no longer used for H210IPZ installation.

H210IPZ Release 7.0

NetEx/IP Release 7.0 provided support for native IP connectivity. This allows an onboard IP connectivity device, such as that provided by an Open Systems Adapter (OSA), to be used as the connection to a NetEx/IP network, in place of using a NESiGate appliance as an ESCON channel gateway to the IP network. In addition to OSA support, H210IPZ also supports NESiGate appliances, to provide interoperability between OSA and NESiGate appliances, as well as with all other NetEx/IP nodes on the network. This interoperability makes it easy to deploy OSA connectivity into an existing NetEx/IP network.

H210IPZ with OSA requires the customer to be licensed for OSA support. A software key, based on CPU serial number and an OSA support feature, is required to enable H210IPZ for OSA.

H210IPZ with NESiGate requires the customer to be licensed for NESiGate-IP support. A software key, based on CPU serial number and an IP support feature, is required to enable H210IPZ for NESiGate ESCON channel interfaces.

H210IPZ can also be licensed for both OSA and NESiGate-IP support.

Please refer to the “Obtaining the Software Key” section on page 42 for the procedure to use for obtaining the key.

Please refer to the “NetEx/IP IP Addressing” section on page 11 for a description of the NetEx/IP IP support.

Note: DXE channel adapters are not supported by H210IPZ.

H210IP Release 6.1

NetEx/IP Release 6.1 provided support for high-performance channel I/O when used in conjunction with the HCM2 channel protocol option of the NESiGate ESCON interface. To use this feature requires the correct version of the NESiGate ESCON channel interface hardware and software. Additionally, the NESiGate channel interface must be configured to operate in HCM2 mode. No additional configuration statements are required in H210IP to support this feature. If the NESiGate device is configured for HCM2, then this version of H210IP will operate those devices with optimized channel programs.

A software key, based on CPU serial number and an IP support feature, is required to enable H210IP for use with NESiGate ESCON channel interfaces.

NetEx Characteristics

NetEx centralizes network considerations in a single piece of software. The following sections describe the characteristics of NetEx:

- External Interface
- Internal Interaction
- NetEx Connections
- Design Efficiency and Flexibility
- Block Segmenting
- Alternate Path Retry
- Remote Operator Interface
- User Exits
- Basic I/O Flow

External Interface

The NetEx external interface for the application programmer is common for all versions of NetEx. NetEx provides requests for use in the programs that call NetEx. These calling programs may be written in FORTRAN, PASCAL, COBOL, C or Assembler languages. NetEx programs written in high-level languages may be transported from one host to another, with some changes to account for different word sizes and other machine architecture variations.

NetEx also provides an operator interface that may be used to monitor and control certain NetEx functions.

Internal Interaction

The internal operations of all supported versions of NetEx are consistent and allow the different versions to interact freely. Thus, any program using NetEx may communicate with any other similar program on the network that is also using NetEx.

To facilitate communication between hosts of different manufacture, NetEx supports character code conversion.

NetEx Connections

To communicate using NetEx, two calling programs first form a connection using a handshake protocol. NetEx then allows this pair of programs to communicate.

NetEx can establish multiple connections at one time, and can allow one program to have multiple connections simultaneously.

NetEx also supports communications within a single host. A calling program may connect to another calling program in the same host and exchange information just as if network communications were taking place.

Design Efficiency and Flexibility

The NetEx design enables many applications on the same processor to share the use of the network facility. Programs calling NetEx can be written without regard to the other programs calling NetEx.

Once NetEx accepts data from the caller, NetEx must deliver the data to its destination. The NetEx subsystem on each host handles flow control, error recovery, and alternate path routing, and is able to tolerate long delays inherent in communication paths over satellite links.

NetEx optimizes data transfer throughput using a high degree of parallelism. That is, under normal circumstances, simultaneous I/O, NetEx buffer management, and user file I/O all take place concurrently. This means that the effective data transfer rate is as fast as possible (in the multi-megabit range).

Block Segmenting

NetEx provides block segmenting at the transport layer. NetEx divides data into segments of a specified size for transmission across the network and reassembles the segments on the remote host before delivering the data to the session layer calling program on the remote NetEx. This segmenting is transparent to the session user, but provides control of the transmitted block segment size. This is especially useful for satellite communication.

Alternate Path Retry

Alternate Path Retry (APR) provides the capability for connections to automatically reroute on different network paths when a failure on a path is detected. This rerouting takes place with no loss of data. For more information on APR, refer to the “C” *Configuration Manager and NetEx Alternate Path Retry (APR) User Guide*.

Remote Operator Interface

NetEx provides a remote operator interface that allows users to issue NetEx operator commands to other defined NetEx hosts on the network. Security features are provided.

User Exits

NetEx provides user exits at well-defined points for security, accounting, or other user-defined purposes. These exits are routines that are essentially do-nothing and may be replaced by user modules at installation time. Because there is a wide variety of user requirements, NetEx does not provide generalized security and accounting needs. User Exits can be implemented to satisfy these requirements.

Basic I/O Flow

The following figure shows the basic I/O flow between two programs using host based NetEx. The calling program communicates with NetEx through the user interface. NetEx then uses the network to communicate with the calling program on the other processor.

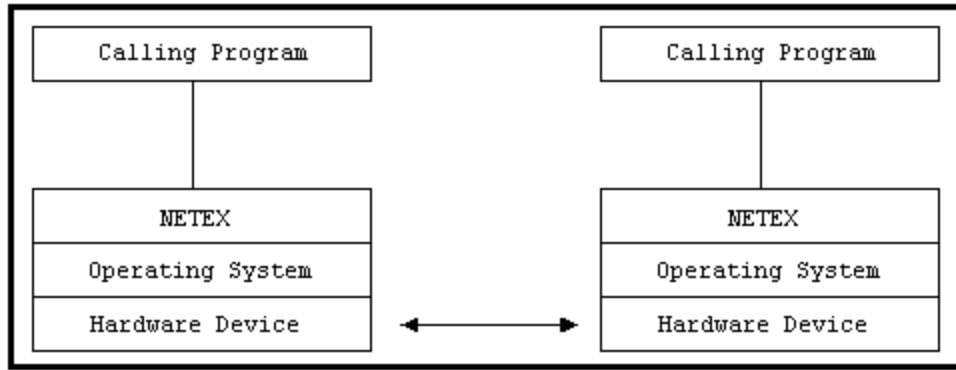


Figure 1. Basic I/O Flow

NetEx and the ISO Model

NetEx follows the guidelines set by the International Standards Organization (ISO) for Open Systems Interconnection (OSI). Open Systems Interconnection refers to the exchange of information among terminal devices, computers, people, and networks, that are open to communication with one another.

The ISO model is composed of seven layers. Each layer interacts only with adjacent layers in the model (see figure). By using this modular structure, the internal function of each layer is self-contained and does not affect the functioning of other layers.

Table 2. ISO Model	
Layer	Major Functions
Application	High level description of data to be transferred and the destination involved
Presentation	Select data formats and syntax
Session	Establish session connection, report exceptions, and select routing
Transport	Manage data transfer and provide NetEx-to-NetEx message delivery
Network	Point-to-point transfer, error detection, and error recovery
Data Link	Data link connection, error checking, and protocols
Physical	Mechanical and electrical protocols and interfaces

Although each layer physically interacts only with adjacent layers, each layer appears to communicate directly with the corresponding layer of the other model. Figure 2 illustrates this concept.

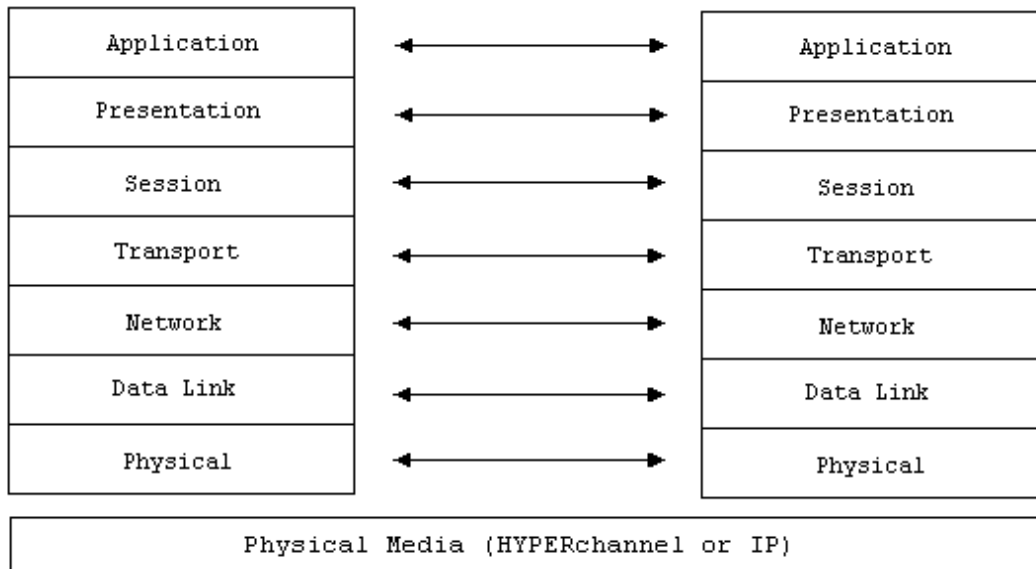


Figure 2. ISO Model Communication

Note: The corresponding layers appear to communicate directly as indicated by the lines with arrows. In practice, they communicate only by progressing down through the layers of one model, through the physical media, and up through the layers of the other model.

The figure below shows the correspondence between NetEx and application software to the ISO model. NetEx software provides complete session, transport, and network layer interfaces. This leaves the user free to write the application programs that use NetEx or to use applications provided by Network Executive Software.

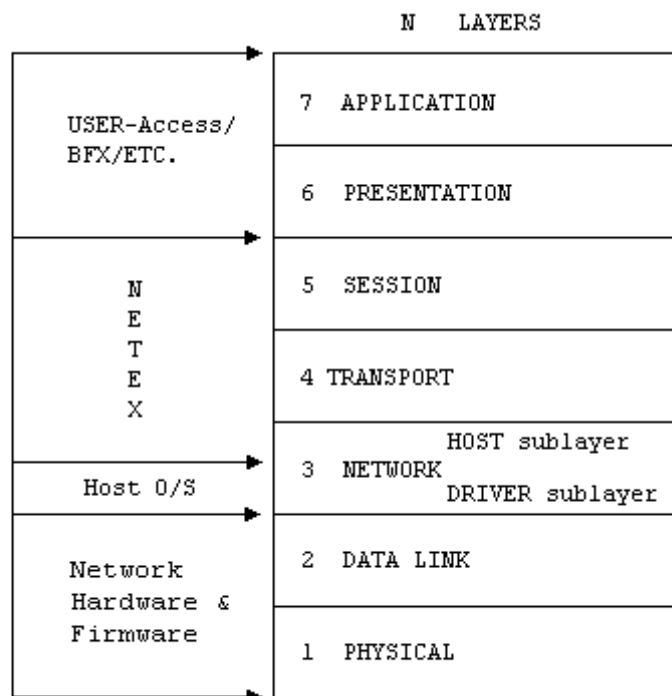


Figure 3. NetEx and the ISO Model

Session Layer Services

As the highest layer within NetEx (referring to the ISO model in Figure 2), the NetEx session layer software provides the general interface to the user's application/utility program. The NetEx session layer services include: program-to-program connection using the best available network path, reading data, writing data, disconnection, and statistics gathering. The user requests these services by using a standard NetEx Request Block (NRB) (containing parameters), and issuing the requests described in the *H210IPZ NetEx/IP for z/OS Operating Systems Programming Reference Manual*. The session layer software processes user requests by requesting services from the underlying transport layer.

Transport Layer Services

The transport layer provides the actual data movement services for NetEx. This is an internal layer used only by the session service code, not the end user. It transmits and receives user data, along with internal protocol information, to provide fast, efficient communications over the network. The transport layer accomplishes its function by performing services for the session layer software above it and by requesting services of the network layer below it.

The transport software manages the network path chosen by the session software. The session user does not need to be concerned with the actual hardware and software used to transmit data, nor with NetEx-to-NetEx message delivery. The transport layer sets up hardware and software tables, provides buffering, and establishes linkages to manage the flow of information. Also, the protocol used by the transport layer software provides true full-duplex communications between subsystems, permitting asynchronous reads and writes. Because the transport layer provides a full-duplex operation, data can flow continuously, as long as data is available. This keeps the communications link as busy as possible and assures timely arrival of data to the user.

Network Layer Services

The network layer software provides link independence for the higher layers of NetEx and assumes responsibility for keeping the network interfaces busy. This is an internal layer used only by the internal transport service, not the end user. The network layer formats the message proper to route the data through the network. If the protocol information overflows the NetEx/IP message proper, the network layer splits the data transmissions into two driver requests. The network layer also multiplexes network connections over common driver connections and manages those driver connections.

Driver Sublayer Services

The driver sublayer software is the interface between the network sublayer and the physical network device. The driver converts network sublayer I/O for a particular network path into a form which is understandable to the devices. The driver delivers and receives network messages and associated data to and from the network adapters. The driver also allows retry and error recovery for network adapters, supports assembly/disassembly, and code conversion options, if these are provided by the adapter type and requested by the user's data mode parameter.

NetEx/IP IP Addressing

Overview

NetEx/IP Release 7.0 (H210IPZ) extended the IP support to include support of OSA-Express for communicating directly over standard IP networks from the S/390 system. H210IPZ is able to communicate with other NetEx/IP hosts over standard IP-based infrastructures without requiring channel connectivity on the S/390 host. No changes are required to NetEx applications, since the NetEx Application Programming Interfaces (APIs) are not changed.

H210IPZ supports the use of OSA devices. . NetEx/IP connections are driven to non-S/390 NetEx/IP servers (e.g. Solaris, Linux, HP, Windows, Unisys, etc) by using an OSA device.

OSA

An Open Systems Adapter (OSA-Express) device provides NetEx/IP with native IP connectivity.

NetEx Network Address

Each NetEx/IP OSA network interface must be assigned a unique NetEx network address (NETADDR and SMGDREF) in the NCT.

NetEx IP-hostname

A NetEx IP-hostname for the NetEx/IP OSA interface must be defined, that consists of a name in the following format:

`NTX0000uu ss`

where ***uu*** is the network address and ***ss*** is the subaddress, as defined in the NCT (NETADDR and SMGDREF respectively). This hostname must resolve to the IP address of the NetEx/IP OSA interface, either through the use of a name server, a local host file, or through the use of the NetEx/IP 'SET IP gna ip' command.

The NetEx/IP OSA IP address must also be defined as a local IP address in the HOME statements for the TCP/IP stack that H210IPZ will use.

An internal NetExAddress-to-IPAddress mapping table is used to route NetEx traffic, identified by *uu ss* in the NetEx network messages, to the correct IP network address. This mapping table is built dynamically.

IP Name Resolution

H210IPZ uses standard name resolution techniques to resolve the NetEx/IP hostnames (i.e. the ntx0000uu ss names). They can be resolved by a DNS server that is pointed to by the TCP/IP stack, or they can be resolved in a local host file that exists on the z/OS system. Alternatively, these host names can also be mapped to their IP addresses with the NetEx/IP SET IP command, or defined in the NetEx/IP defaults file.

Virtual IP Address (VIPA) Support

NetEx/IP can support either static or dynamic VIPA. The NetEx/IP GNA Network Address can map to the IP address of either a real physical interface, or a Virtual IP Address (VIPA).

Refer to IBM's *z/OS Communications Server IP Configuration Guide* for descriptions of VIPA configuration requirements.

OSA NCT Requirement

OSA devices are defined in the Network Configuration Table (NCT) with an ADAPTER definition. Refer to "OSA NCT Considerations" on page 65 for a description of the OSA NCT requirements.

OSA NTXINIT Initialization File Requirement

The OSAA initialization statement in the NTXINIT initialization file identifies local ADAPTER definitions in the NCT as OSA devices, and optionally specifies the IP address that NetEx/IP should bind to during initialization. One OSAA statement must be specified for each OSA interface being used. OSAA statements should only be used to define local OSA adapters, not remote OSA adapters.

The OSAA statement identifies the NetEx/IP OSA adapter by specifying it in either of the following formats:

1. Using only the NETADDR portion of the OSA ADAPTER NCT definition. This format is provided for compatibility with the format used in H210IPZ Release 7.0. For example:

OSAA 03

2. Using both the NETADDR and SMGDREF portions of the OSA ADAPTER NCT definition. For example:

OSAA 0300

If the second format is used, then an optional IP address can also be specified on the OSAA statement, to identify the local IP address that NetEx/IP should bind to during initialization. This IP address can either be the IP address of a real physical interface, or it can be a virtual (VIPA) address. For example:

OSAA 0300 10.1.10.55

Sample OSAA statements are contained in the sample NTXINIT initialization file located in hlq.NTXPARM(NTXPARM).

NetEx/IP Pre-Installation

General

This section contains pre-installation information and requirements pertaining to the various configurations supported by H210IPZ. This section contains information on the following topics:

- Prerequisites
- z/OS Considerations
- OSA Configuration Requirements
- Alternate Path Retry
- TNP

Prerequisites

The H210IPZ NetEx/IP software product runs on z/OS operating systems, and uses an OSA-Express interface as a network device for direct access to the IP network. The prerequisites to run H210IPZ NetEx/IP with OSA-Express are as follows:

- A software key from Network Executive Software that enables OSA-Express support.
- An IBM (or compatible) processor running z/OS.
- A minimum of three OSA devices, configured and available on at least one OSA-Express interface.
 - Multiple OSA interfaces can be used by NetEx/IP
 - VIPA configurations are supported
- z/OS Communications Server must be properly configured for NetEx/IP and active.
- VTAM must be properly configured for NetEx/IP and active.
- RACF (or other security product) must be configured to authorize NETEX
- If Policy Agent is used, it must be configured to authorize NETEX
- If other servers on the network intend to communicate with applications on this host through NetEx/IP, they must have NetEx/IP installed and active.

H210IPZ does not require any other software products to run in a z/OS environment, other than the base operating system and the linkage editor or binder.

z/OS Considerations

H210IPZ can use OSA adapters as either local or remote NetEx/IP devices. For local devices, H210IPZ will use socket calls for local OSA interfaces. NetEx/IP traffic is directed to the remote NetEx/IP Network Addresses, which can be remote H210IPZ OSA interfaces or other remote Unisys, Linux/Unix/Windows NetEx/IP hosts.

Refer to the “OSA Configuration Requirements” section on page 18 for a description of the z/OS Configuration Requirements for using OSA interfaces.

ALLOWUSERKEYCSA Support

NetEx/IP requires a small amount of CSA memory. To run NetEx/IP on a system on which ALLOWUSERKEYCSA has been specified (or defaulted) as NO, a z/OS Program Properties Table (PPT) entry must be defined for program name NETEX that specifies that NetEx/IP runs in a system key (2-7).

This can be accomplished by creating a ‘SCHEDxx’ PARMLIB member that contains the following entry, where n specifies a value between 2-7:

<pre>PPT PGMNAME (NETEX) KEY (n)</pre>

Figure 4. ALLOWUSERKEYCSA SCHEDxx member

The new SCHEDxx member can be dynamically activated with the SET command (SET (or T) SCH=xx). To make this change persistent, modify the ‘SCH=xx’ parameter in the current IEASYSy PARMLIB member to point to the new SCHEDxx member. This change will then become persistent when the next IPL is performed

REUSASID Support

Release 7.4 supports the specification of either REUSASID(YES) or REUSASID(NO) in the DIAGxx PARMLIB member. If REUSASID(YES) is specified, NetEx/IP can optionally be run in a reusable address space by starting NetEx/IP with the REUSASID=YES parameter as follows:

```
S NETEX,REUSASID=YES
```

SMF Interface

H210IPZ NetEx can be configured to create SMF records which describe NetEx session activity. This feature uses the H210IPZ DISCONNECT user exit. During exit processing, NetEx copies information available to the DISCONNECT exit into the SMF DSECT named “SMFDSECT” within module NXMUX5, it then passes it onto SMF via the SMFWTM macro.

The base release includes a working SMF exit example in module NXMUX5. This example shows how to use the DISCONNECT exit and generate SMF records. You can modify this example to include any information available with the DISCONNECT exit.

SMF Installation

H210IPZ SMF recording uses the DISCONNECT user exit NXMUX5. This section describes which NetEx parameters must be set to enable the generation of SMF records.

Set the following parameters in the NetEx initialization parameter file (ddname NTXINIT) before you start NetEx:

SMFTYPE

Establishes the SMF record identifier number that NetEx uses when generating SMF records.

RECORDTYPE

Specify a value between 0 and 255. Note that record types 128-255 are reserved for user written records. There is no default.

XDISC

Enables/disables the NetEx DISCONNECT user exit.

ON

Enables the NetEx DISCONNECT user exit.

OFF

Disables the NetEx DISCONNECT user exit. The default is OFF.

IOCOUNT

Specifies the recording of I/O activity events.

ON

Enables recording of I/O activity events.

OFF

Disables recording of I/O activity events. The default is OFF.

Example

The following example shows how to define the parameters to create SMF records:

```
SMFTYPE=128
XDISC=ON
IOCOUNT=ON
```

Related Topics

See the following sections for more information on SMF record reporting:

- “SMF Record Layout” on page 17
- “IOCOUNT” on page 96
- “SMFTYPE” on page 125
- “XDISC” on page 145

SMF Record Layout

Table 3. SMF Record Layout				
Offset	Label	Length	Format	Description
00	SMFREC	88	Binary	NetEx SMF Record
00	SMFREC	18	Binary	SMF Header
00	SMFLENG	2	Binary	Record Length
02	SMFSUBR	2	Binary	Segment Descriptor
04	SMFFLAG	1	Binary	System Indicator
05	SMFTYPE	1	Binary	Record Type
06	SMFTIME	4	Binary	Record Create Time
0A	SMFDATE	4	Packed	Record Create Date
0E	SMFSYST	4	EBCDIC	System Identification
12	SMFDATA	70	Binary	NetEx Data
12	XITUSER	8	EBCDIC	User/Owner ID
1A	XRBPNAME	8	EBCDIC	Process Name
22	XRBNHNAME	8	EBCDIC	Host Name
2A	XRBLLEN	4	Binary	NRB Length
2E	XRBBLKO	4	Binary	Output BLKSIZE
32	XTSNRDC	4	Binary	Session Reads
36	XTSNWRC	4	Binary	Session Writes
3A	XTTRXMT	4	Binary	Session Retransmits
3E	XTNWRTS	4	Binary	Network Writes
42	XTNRDS	4	Binary	Network Reads
46	XCXWADAP	4	EBCDIC	Adapter Definition
4A	XCXWAPCN	4	Binary	PDATA Count
4E	XCXWAOCN	4	Binary	ODATA Count
52	XRBMXRT	4	Binary	Maximum Transmit Rate
56	SMFDSEND			End of SMF Record

OSA Configuration Requirements

H210IPZ was designed to use an OSA-Express Multipath Channel (MPC) device in Queued Direct I/O (QDIO) mode. OSA-Express devices must be defined to the operating system with HCD or IOCP, as well as be configured in VTAM and TCP/IP before they can be used by NetEx/IP.

The process of defining OSA channel paths, control units and devices is very similar to the process of defining other types of devices. An OSA-Express channel path is defined as an **OSD channel**, an OSA-Express control unit is defined as an **OSA control unit type** on the **OSD channel**, and OSA-Express devices are defined as **OSA devices**, connected to the **OSA control unit type**, which is on the **OSD channel**.

The OSA-Express devices used by NetEx/IP are not required to be exclusive to NetEx/IP. NetEx/IP is merely another IP application that can use these shared OSA devices.

The OSA-Express devices must also be defined to VTAM and TCP/IP before they can be used by any IP application. The following overview is only meant to represent a high-level example. Refer to the following IBM publications for more specific rules and requirements when defining OSA-Express devices:

z/OS Communications Server: IP Configuration Guide

z/OS Communications Server: SNA Resource Definition Reference

z9 and zSeries OSA-Express Customer's Guide and Reference

CNTLUNIT Statement

An OSA controller must be defined as an **OSA** control unit type.

IODEVICE Statement

OSA devices must be defined as an **OSA** device type.

Device I/O Definition Requirements

The device type used for defining the NetEx devices is dependent on the particular device being used.

OSA-Express Device

If an OSA-Express interface is used, it must be defined as an OSA device type.

Note: The use of an OSA-Express interface requires a software key that enables OSA support.

NetEx/IP requires a minimum of three OSA devices on the OSA controller. However, these devices are not exclusive, and may be shared with other IP applications.

OSA Device Definitions

Figure 5 on page 19 shows the channel, control unit, and device definition requirements for defining OSA-Express NetEx/IP devices, along with an indication of support for DYNAMIC device definitions, the LOCANY parameter, and EMIF. Either IOCP or HCD can be used when defining the devices.

CHPID Type	Control Unit Type	Device Type	Device Features Supported		
			DYNAMIC	LOCANY	EMIF
OSD	OSA	OSA	YES	YES	YES

Figure 5. NetEx/IP OSA-Express device definition requirements

VTAM Considerations

NetEx/IP OSA-Express devices must be defined to VTAM in a Transport Resource List Element (TRLE) in a Transport Resource List (TRL) major node. TRL definitions are located in SYS1.VTAMLST, or a customer hlq.VTAMLST library. This node must be activated to VTAM before the OSA-Express devices can be used by TCP/IP. An example of defining OSA-Express devices as part of a TRL major node definition is shown in Figure 6.

```

TRL          VBUILD TYPE=TRL
*
*
TRLD00A TRLE LNCTL=MPC,MPCLEVEL=QDIO,READ=F000,WRITE=F001,      X
              DATAPATH=( F002 ),PORTNAME=OSDCHP01

```

Figure 6. Sample TRLE Definition

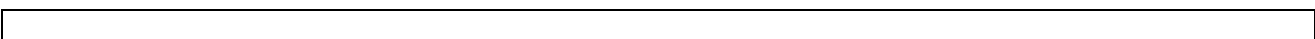
One transport resource list element (TRLE) must be defined for each multipath channel (MPC) connection. The LNCTL operand indicates that the link is a multipath channel link. The MPCLEVEL operand indicates that the Queued Direct I/O interface is used.

The READ, WRITE, and DATAPATH operands specify the OSA devices that were defined with HCD. The READ and WRITE devices are dedicated ‘control’ devices, while the DATAPATH device specifies a device used for sending and receiving data through the OSA-Express interface. The READ operand must specify only one device, and that device must be an even number that is one less than the corresponding WRITE operand. The WRITE operand must specify only one device, and that device must be an odd number that is one greater than the corresponding READ operand. The DATAPATH operand may specify multiple devices. However, only one device is used for each instance of TCP/IP in the same logical partition.

The name specified on the PORTNAME parameter must be the same name specified for the device name on the corresponding TCP/IP DEVICE statement.

TCP/IP Considerations

Device and Link definitions are contained in the TCP/IP configuration file (e.g. TCPIP.PROFILE.TCPIP). These statements define the device to TCP/IP. The name specified on the DEVICE statement (e.g. OSDCHP01) must be the same name specified for the PORTNAME in the corresponding VTAM TRLE definition. An example of defining the OSA-Express device from Figure 6 in the TCPIP.PROFILE.TCPIP data set is shown in Figure 7.



```

; DEVICE and LINK for MPC OSA Ethernet Devices
;
; OSA on subnet 6 (10.1.6.11)
DEVICE   OSDCHP01      MPCIPA      NONROUTER  AUTORESTART
LINK     OSACHP01LK    IPAQENET   OSDCHP01   READSTORAGE MAX

```

Figure 7. Sample TCP/IP OSA-Express Definition

Note that the DEVICE name (OSDCHP01) is the same name specified as the PORTNAME in the TRLE definition in Figure 6.

Be sure to also include a START command in the TCPIP.PROFILE.TCPIP dataset (e.g. START OSDCHP01) for the OSA device added. TCP/IP must either be restarted or refreshed with the new

TCPIP.PROFILE.TCPIP

dataset before the new OSA device can be used.

Refer to the “*z/OS Communications Server: IP Configuration Guide*” for a description of the DEVICE and LINK operands.

RACF Considerations

The user name associated with the NETEX started task must have an OMVS segment defined. If it does not, the Netex BIND will fail during initialization, and a Netex SMI003 abend will occur. Since there are various ways this can be set up in RACF, the specific RACF commands to do this are dependent on how you administer your security policies. You may need to work with your Security Administrator, but the basic requirement is to ensure the user name associated with the Netex started task has an OMVS segment defined.

Policy Agent Considerations

If z/OS Policy Agent is being used, determine if policy changes are needed in order to authorize NetEx/IP access over the IP network. Refer to “Step 15. (Optional) Update Policy Agent” on page 60 in the installation instructions for NetEx/IP Policy Agent requirements.

OSA NetEx/IP Configuration Examples

NetEx/IP on z/OS supports both Static and Dynamic VIPA configurations, as well as non-VIPA configurations.

Note: *The examples in this section are not meant to convey the full extent of VTAM and TCP/IP parameters, but only to indicate the general concepts and relationships of how the NetEx/IP addressing works in various IP configurations.*

Refer to the following IBM manuals for full descriptions of TCP/IP configuration parameters and requirements:

z/OS Communications Server IP Configuration Guide

Examples of NetEx/IP Non-VIPA configurations

In these examples, NetEx/IP uses one or more IP addresses that represent IP addresses of real physical ethernet interfaces.

Example 1: One physical ethernet interface

A sample portion of the VTAMLST TRLE definitions (e.g. contained in SYS1.VTAMLST or equivalent) is shown in Figure 8. A TRLE defines the physical devices and portname associated with each OSA adapter.

```
*
TRL          VBUILD TYPE=TRL
*
*
TRLD00A      TRLE  LNCTL=MPC ,                                X
                MPCLEVEL=QDIO ,                              X
                READ=( F000 ) ,                                X
                WRITE=( F001 ) ,                               X
                DATAPATH=( F002 ) ,                             X
                PORTNAME=DCHP01
*
```

Figure 8. Sample TRLE entry with 1 OSA

A sample portion of the TCP/IP Profile dataset (e.g. contained in TCPIP.PROFILE.TCPIP or equivalent) is shown in Figure 9. The HOME statement defines 10.1.6.120 as the IP address of the real ethernet interface.

```

;
; DEVICE and LINK for MPC OSA Ethernet Devices
; OSA on subnet 6 (10.1.6.120)
;
DEVICE    DCHP01  MPCIPA          NONROUTER  AUTORESTART
LINK      DCHP01LK      IPAQENET    DCHP01      READSTORAGE MAX
;
;HOME IP address for each link in the host
;
HOME 10.1.6.120 DCHP01LK          ; OSA Card CHP(01)
;

```

Figure 9. Sample TCP/IP Profile with 1 OSA

A sample portion of the NetEx/IP NCT that defines the OSA Adapter is shown in Figure 10. Note that this looks no different than any other NetEx/IP Adapter definition.

```

*
ZOS2      HOST    TYPE=IBM  Model=Z9
          PROTOCOL=2
*
* z/OS NetEx/IP OSA at NetEx/IP network address 0300
*
          ADAPTER MODEL=N220  NETADDR=03  CHANADDR=0300
          T0=ETH1  NUMADDRES=2  SMGDREF=00
*

```

Figure 10. Sample NetEx/IP NCT with 1 OSA

A sample portion of the NetEx/IP Initialization Parameters (NTXINIT) that identifies Adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA device is shown in Figure 11. "OSA1IF ON" indicates that NetEx/IP will BIND to address 0 during NetEx/IP initialization, regardless of any DNS name resolution.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*
OSAA 0300
*
*** OSA1IF ON indicates that NetEx/IP will BIND to 0
*
OSA1IF ON
*

```

Figure 11. Sample NetEx/IP NTXINIT with 1 OSA and OSA1IF ON

An alternative solution for the NetEx/IP Initialization Parameters is shown in Figure 12. In this case, specifying "OSA1IF OFF" indicates that NetEx/IP will BIND to a specific IP address during NetEx/IP initialization. For this case to work properly, host name ntx00000300 must resolve (e.g. by DNS lookup) to an IP address that NetEx/IP can use for the BIND.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*
OSAA 0300
*
*** OSA1IF OFF indicates that NetEx/IP will BIND to a specific IP address
***      (obtained by name resolution)
*
OSA1IF OFF
*

```

Figure 12. Sample NetEx/IP NTXINIT with 1 OSA and OSA1IF off

Another alternative solution for the NetEx/IP Initialization Parameters is shown in Figure 13. In this case, specifying an IP address on the OSAA statement indicates that NetEx/IP will BIND to that specific IP address during NetEx/IP initialization without requiring the services of DNS. The “OSA1IF” statement is ignored, but should be specified as OFF for clarity of documentation.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*
OSAA 0300 10.1.6.120
*
*** OSA1IF is ignored when the IP address is explicitly specified
*** on the OSAA statement.
*
OSA1IF OFF
*

```

Figure 13. Sample NetEx/IP NTXINIT with 1 OSA and OSAA defines IP

Example 2: Two physical ethernet interfaces

A sample portion of the VTAMLST TRLE definitions (e.g contained in SYS1.VTAMLST or equivalent) is shown in Figure 14. Each TRLE defines the physical devices and portname associated with each OSA adapter.

```

*
TRL          VBUILD TYPE=TRL
*
*
TRLD00A      TRLE  LNCTL=MPC,                                X
                MPCLEVEL=QDIO,                                X
                READ=(F000),                                    X
                WRITE=(F001),                                    X
                DATAPATH=(F002),                                X
                PORTNAME=DCHP01
*
TRLD00B      TRLE  LNCTL=MPC,                                X
                MPCLEVEL=QDIO,                                X
                READ=(E000),                                    X
                WRITE=(E001),                                    X
                DATAPATH=(E002),                                X
                PORTNAME=DCHP31
*

```

Figure 14. Sample TRLE with 2 OSAs

A sample portion of the TCP/IP Profile dataset (e.g. contained in TCPIP.PROFILE.TCPIP or equivalent) is shown in Figure 15. The HOME statements define 10.1.6.120 and 10.1.5.82 as the IP addresses of the real ethernet interfaces.

```

;
; DEVICE and LINK for MPC OSA Ethernet Device DCHP01
; OSA on subnet 6 (10.1.6.120)
;
DEVICE  DCHP01  MPCIPA          NONROUTER  AUTORESTART
LINK    DCHP01LK          IPAQENET  DCHP01          READSTORAGE MAX
;
; DEVICE and LINK for MPC OSA Ethernet Device DCHP31
; OSA on subnet 5 (10.1.5.82)
;
DEVICE  DCHP31  MPCIPA          NONROUTER  AUTORESTART
LINK    DCHP31LK          IPAQENET  DCHP31          READSTORAGE MAX
*
;
;HOME IP address for each link in the host
;
HOME 10.1.6.120 DCHP01LK          ; OSA Card CHP(01)
HOME 10.1.5.82  DCHP31LK          ; OSA Card CHP(31)
;

```

Figure 15. Sample TCP/IP Profile with 2 OSAs

A sample portion of the NetEx/IP NCT that defines the OSA Adapters is shown in Figure 16. Note that these look no different than any other NetEx/IP Adapter definitions.

```

ZOS2      HOST  TYPE=IBM  Model=Z9
          PROTOCOL=2
*
* z/OS NetEx/IP OSA at NetEx/IP network address 0300
*
          ADAPTER MODEL=N220  NETADDR=03  CHANADDR=0300
          T0=ETH1  NUMADDRS=2  SMGDREF=00
*
* z/OS NetEx/IP OSA at NetEx/IP network address 0540
*
          ADAPTER MODEL=N220  NETADDR=05  CHANADDR=0540
          T0=ETH1  NUMADDRS=2  SMGDREF=40
*

```

Figure 16. Sample NetEx/IP NCT with 2 OSAs

Figure 17 is an example of an invalid configuration, since OSA1IF should never be set to ON when there is more than 1 OSAA statement. NetEx/IP will successfully BIND to 0 for the first OSAA processed, but will fail when trying to BIND to 0 for any subsequent OSAA statements.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*** NCT adapter 0540 (NETADDR 05, SMGDREF 40) as an OSA Adapter
*
OSAA 0300
OSAA 0540
*
*** OSA1IF ON indicates that NetEx/IP will BIND to 0
*** This is an invalid setting when there is more than one OSAA statement
*
OSA1IF ON
*

```

Figure 17. Sample NetEx/IP NTXINIT with 2 OSAs and OSA1IF ON – Invalid configuration

A correct solution for the NetEx/IP Initialization Parameters is shown in Figure 18. In this case, specifying “OSA1IF OFF” indicates that NetEx/IP will BIND to a specific IP address during NetEx/IP initialization. For this case to work properly, host names ntx00000300 and ntx00000540 must both resolve (e.g. by DNS lookup) to IP addresses that NetEx/IP can use for each specific BIND.


```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*** NCT adapter 0540 (NETADDR 05, SMGDREF 40) as an OSA Adapter
*
OSAA 0300
OSAA 0540
*
*** OSA1IF OFF indicates that NetEx/IP will BIND to a specific IP address
***      (obtained by name resolution)
*
OSA1IF OFF
*

```

Figure 18. Sample NetEx/IP NTXINIT with 2 OSAs and OSA1IF OFF

An alternative solution for the NetEx/IP Initialization Parameters is shown in Figure 19. In this case, specifying IP addresses on the OSAA statements indicate that NetEx/IP will BIND to each of these specific IP address during NetEx/IP initialization without requiring the services of DNS. The “OSA1IF” statement is ignored, but should be specified as OFF for clarity of documentation. Note: if an IP address is specified on any OSAA statement, an IP address must be specified on all OSAA statements.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*** NCT adapter 0540 (NETADDR 05, SMGDREF 40) as an OSA Adapter
*
OSAA 0300 10.1.6.120
OSAA 0540 10.1.5.98
*
*** OSA1IF is ignored when the IP address is explicitly specified
*** on the OSAA statement.
*
OSA1IF OFF
*

```

Figure 19. Sample NetEx/IP NTXINIT with 2 OSAs and OSAA defines IP

Example of VIPA configuration

In this configuration, two IP addresses (10.1.6.120 and 10.1.5.82) represent two real physical ethernet interfaces, and a VIPA device is defined at 10.1.8.50.

There are no special parameters for configuring VIPA in NetEx/IP. For this configuration, NetEx/IP should be configured to use the VIPA device. This can be accomplished by defining one Adapter in the NetEx/IP NCT to represent the VIPA device, and identifying it as OSA with the OSAA statement in the NetEx/IP initialization file. The IP address of the VIPA device can either be specified on the OSAA statement, or the NetEx/IP “hostname” for that Adapter must resolve to the VIPA IP address. The NetEx/IP “hostname” consists of *ntx0000uu*, where “uu” and “ss” are the NETADDR (uu) and SMGDREF (ss) of the OSA Adapter in the NCT.

A sample portion of the VTAMLST TRLE definitions (e.g. contained in SYS1.VTAMLST or equivalent) is shown in Figure 20. Each TRLE defines the physical devices and portname associated with each OSA adapter. There are no TRLE entries for VIPA devices.

```

*
TRL          VBUILD TYPE=TRL
*
*
TRLD00A      TRLE  LNCTL=MPC ,                                X
                  MPCLEVEL=QDIO ,                             X
                  READ=( F000 ) ,                             X
                  WRITE=( F001 ) ,                             X
                  DATAPATH=( F002 ) ,                           X
                  PORTNAME=DCHP01
*
TRLD00B      TRLE  LNCTL=MPC ,                                X
                  MPCLEVEL=QDIO ,                             X
                  READ=( E000 ) ,                             X
                  WRITE=( E001 ) ,                             X
                  DATAPATH=( E002 ) ,                           X
                  PORTNAME=DCHP31
*

```

Figure 20. Sample TRLE with 2 OSAs and VIPA

A sample portion of the TCP/IP Profile dataset (e.g. contained in TCPIP.PROFILE.TCPIP or equivalent) is shown in Figure 21. The HOME statements define 10.1.8.50 as the VIPA address, and 10.1.6.120 and 10.1.5.82 as the IP addresses of the real IP ethernet interfaces.

```

;
; DEVICE and LINK for Virtual Device (VIPA)
;
DEVICE VIPADV    VIRTUAL    0          ; Static VIPA device
LINK    VIPADV LK VIRTUAL    0    VIPADV
;
; DEVICE and LINK for MPC OSA Ethernet Device DCHP01
; OSA on subnet 6 (10.1.6.120)
;
DEVICE    DCHP01 MPCIPA          NONROUTER    AUTORESTART
LINK      DCHP01LK          IPAQENET    DCHP01          READSTORAGE MAX
;
; DEVICE and LINK for MPC OSA Ethernet Device DCHP31
; OSA on subnet 5 (10.1.5.82)
;
DEVICE    DCHP31 MPCIPA          NONROUTER    AUTORESTART
LINK      DCHP31LK          IPAQENET    DCHP31          READSTORAGE MAX
*
;
;HOME IP address for each link in the host
;
HOME 10.1.8.50    VIPDV LK          ; VIPA for Stack
HOME 10.1.6.120   DCHP01LK          ; OSA Card CHP(01)
HOME 10.1.5.82    DCHP31LK          ; OSA Card CHP(31)
;

```

Figure 21. Sample TCP/IP Profile with 2 OSAs and VIPA

A sample portion of the NetEx/IP NCT that defines the Virtual OSA Adapter is shown in Figure 22. Note that this looks no different than any other NetEx/IP Adapter definition.

```

*
ZOS2    HOST    TYPE=IBM    Model=Z9
          PROTOCOL=2
*
* z/OS NetEx/IP VIPA OSA at NetEx/IP network address 0300
*
          ADAPTER MODEL=N220    NETADDR=03    CHANADDR=0300
          T0=ETH1    NUMADDRES=2    SMGDREF=00
*

```

Figure 22. Sample NetEx/IP NCT with 1 VIPA OSA

A sample portion of the NetEx/IP Initialization Parameters (NTXINIT) that identifies Adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA device, is shown in Figure 23. "OSA1IF ON" indicates that NetEx/IP will BIND to address 0 during NetEx/IP initialization.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*
OSAA 0300
*
*** OSA1IF ON indicates that NetEx/IP will BIND to 0
*
OSA1IF ON
*

```

Figure 23. Sample NetEx/IP NTXINIT with 1 VIPA OSA and OSA1IF ON

An alternative solution for the NetEx/IP Initialization Parameters is shown in Figure 24. In this case, specifying “OSA1IF OFF” indicates that NetEx/IP will BIND to a specific IP address during NetEx/IP initialization. For this case to work properly, host name ntx00000300 must resolve (e.g. by DNS lookup) to an IP address that NetEx/IP can use for the specific BIND.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*
OSAA 0300
*
*** OSA1IF OFF indicates that NetEx/IP will BIND to a specific IP address
*** (obtained by name resolution)
*
OSA1IF OFF
*

```

Figure 24. Sample NetEx/IP NTXINIT with 1 OSA and OSA1IF OFF

Another alternative solution for the NetEx/IP Initialization Parameters is shown in Figure 25. In this case, specifying an IP address on the OSAA statement indicates that NetEx/IP will BIND to that specific IP address during NetEx/IP initialization without requiring the services of DNS. The “OSA1IF” statement is ignored, but should be specified as OFF for clarity of documentation.

```

*
*** the following initialization statement identifies
*** NCT adapter 0300 (NETADDR 03, SMGDREF 00) as an OSA Adapter
*** and uses the VIPA address for the NetEx/IP BIND
*
OSAA 0300 10.1.8.50
*
*** OSA1IF is ignored when the IP address is explicitly specified
*** on the OSAA statement.
*
OSA1IF OFF
*

```

Figure 25. Sample NetEx/IP NTXINIT with 1 OSA and OSAA defines IP

Dynamic VIPA configuration

A dynamic VIPA can be moved to a different LPAR or system (i.e. it can be moved to a different TCP/IP stack).

There is no special configuration requirement for NetEx/IP to support Dynamic VIPA. NetEx/IP with Dynamic VIPA should be configured the same way as it is for Static VIPA. As long as the NetEx/IP OSA “hostname” (*ntx0000uuss*) resolves to the same VIPA as on the originating system (or is specified on the OSAA initialization statement), the NetEx/IP NCT will remain functional. NetEx/IP can be stopped on the originating system, the VIPA address moved to the new system, then NetEx/IP can be started on the new system.

However, if NetEx/IP is not licensed on the new system, a new software key will have to be obtained. If the usage on the new system is temporary, (e.g. in support of recovering from a system failure), a DR key can be obtained by contacting support@netex.com. A DR key is functional for a 10 day period.

Alternate Path Retry

Alternate Path Retry (APR) automatically reroutes connections to a different network path when NetEx detects a failure on a path. This rerouting occurs with no loss of data. NetEx/IP provides Alternate Path Retry as part of the type 2 and type 4 protocols (as specified on the Configuration Manager HOST statements).

APR is supported for OSA paths.

There are two conditions under which alternate path retry is invoked:

- A connect timeout is reached on the host attempting a connection (connect timeout is set at initialization and may be changed by the operator).
- A dead timeout, after session establishment during normal data delivery, is encountered (dead timeout is set at initialization and may be changed by the operator).

Both these situations can arise because of a failure somewhere on the network.

If a session is already in progress and a loss of communications is detected by the NetEx Transport layer, NetEx attempts to deliver the data to the remote host through an alternate route. This rerouting of data is transparent (excepting the associated transmission delay) to transport and session layer users.

When the transport layer detects a time-out during a connect or data-transfer operation, the timer values are zeroed and an attempt is made to connect to the remote program using the next alternate path. This timeout process continues until either a successful connection is made or until all paths have been tried. If any connection is established, all paths are retried if the connection failed. If NetEx cycles through all alternate paths without a successful connection, a Disconnect Indication is returned to the caller.

The alternate path retry function is invoked by providing more than one route between 2 hosts using the NTCROUTE statement, or Configuration Manager HOST statements. No other action is necessary.

NOTE: Both hosts must be at a minimum level of 3.x NetEx and type 2 or type 4 protocol must be used on this route.

Alternate Path Retry Programming Considerations

When using Alternate Path Retry, the application programmer must carefully choose the read timeout values selected in READ requests. This read timeout value must be a long enough time period to allow NetEx to reach DEADTIME on each alternate path. As a result, the read timeout value specified in the READ requests should be greater than DEADTIME plus connect time (CONNTIME) times the number of alternate paths.

NOTE: DEADTIME and CONNTIME are set in the Initialization file or by the operator.

TNP

The TNP feature serves as a NetEx/IP “proxy”, which can be used by other hosts for which a host-based NetEx/IP is not available. A NetEx/IP Requester, residing on an HP NonStop, OpenVMS, Bull, or Stratus server, works with BFX, USER-Access, eFT, PFX (or other NetEx/IP applications running on the Requester server), and reads and writes the application’s NetEx/IP requests over a TCP/IP connection to TNP on z/OS, which then passes the request to z/OS NetEx/IP, on behalf of the Requester’s NetEx/IP application. In effect, TNP serves as a “proxy” NetEx/IP application for the Requester’s NetEx/IP applications.

Prior to H210IPZ with TNP, Requester applications used a NESiGate-LO device to provide NetEx/IP services. H210IPZ with TNP eliminates the need for NESiGate-LO.

Note: the TNP feature requires OSA as the only network connectivity option. TNP is not supported in NESiGate channel configurations.

TNP is a separately licensed feature of H210IPZ. If enabled, a TNP NetEx/IP application is started during NetEx/IP initialization, and is ready to accept connections from NetEx/IP Requester applications running on different servers. When a NetEx/IP request comes in from a Requester, TNP becomes the local NetEx/IP application, acting on behalf of the Requester. TNP uses the NetEx/IP API’s to communicate with the local NetEx/IP (H210IPZ), just like any other local NetEx/IP applicaion. The Requester application is able to esablish NetEx/IP sessions with other remote NetEx/IP hosts on the network, as well as with H210IPZ. Connections established between Requester applications and H210IPZ effectively appear as NetEx/IP intrahost connections.

The TNP license also contains a session limit count, which places an upper limit on the number of concurrent TNP connections. The TNP license is scaled for different session limit counts.

When running Netex TNP applications from Netex Requesters, it is recommended that a NetEx/IP Requester host entry be added to each of the host-based NetEx/IP NCT’s, and that the multihost feature be enabled in the H210IPZ that is supporting TNP. This can be done by specifying MULTIHOST ON in the NTXINIT initialization file, or by issuing the SET MULTIHOST ON NetEx/IP command. To ensure this setting is persistent across NetEx/IP restarts, the MULTIHOST statement should be added to the initialization file. Refer to the “Adding NetEx/IP Requester Hosts to the NCT” section on page 36 for more information on the NCT requirement.

Figure 26 shows a display of a TNP requester BFXJS application, submitted on host TNPFLASH (a NetEx/IP Requester system). Unique Requester user names are assigned by TNP. They are prefixed with ‘TNP’, followed by 5 numerics.

Host	ZOST	Active Sessions						
Sref	User	Tref	State	Name	Host	RNref	Msg In	Msg Out
1	NTXOPER		OFFERED	NTXOPER	ZOST			
4	TNP		OFFERED	TNPOFFER	ZOST			
5	TNP00001		OFFERED	BFXJS	TNPFLASH			

Figure 26. Output display of an OFFERED TNP job

Figure 27 shows a display of several active TNP connections. Since these are all connections between TNPFLASH and ZOST, these are effectively all intrahost NetEx/IP connections on ZOST, but transferring files between the NetEx/IP Requester host and z/OS.

Host ZOST		Active Sessions							
Sref	User	Tref	State	Name	Host	RNref	Msg In	Msg Out	
1	NTXOPER		OFFERED	NTXOPER	ZOST				
4	TNP		OFFERED	TNPOFFER	ZOST				
5	BFXJST		OFFERED	BFXJST	ZOST				
87	TNP00002	168	CONNECTD	NTXGENDC	ZOST		4901	0	
88	NTXGENDC	166	CONNECTD	TNP00002	ZOST	126	1	4911	
94	BFXTNPFL	184	CONNECTD	TNP00001	ZOST		2	121	
95	TNP00001		OFFERED	BFXJS	TNPFLASH				
96	TNP00001	182	CONNECTD	BFXTNPFL	ZOST	138	110	1	

Figure 27. Output display of connected TNP jobs

TNP Configuration

H210IPZ Configuration

TNP License Key

The H210IPZ software key must indicate that TNP is a supported feature.

TNP JOB or STARTED TASK

During H210IPZ installation, members for TNPJOB, TNPCFG, and optionally TNPPROC were built and/or customized in the installation library “&hlq.NTXCTL”. However, these files can also be edited and changed, if needed, anytime after the H210IPZ installation. If TNP is enabled (via software key), then the job specified by the TNPJOB installation parameter is submitted during NetEx/IP initialization. If TNPSTART(JOB) is specified during H210IPZ installation, the JCL that is customized for TNPJOB is used to run TNP as a batch job.

If TNPSTART(STCSTART) is specified during H210IPZ installation, the JCL that is customized for TNPJOB is used to run TNP as a started task. The job issues the z/OS START TNPPROC command (where TNPPROC is the name specified by the TNPPROC(name) installation parameter, then the batch job terminates.

NOTE: If the TNPSTART(STCSTART) option is specified, z/OS message IEF166D may be displayed when the ‘S tnpprocname’ command is issued, which requires a Y(es) or N(o) authorization response. If the response to message IEF166D is Y, TNP will be allowed to continue; if the response to message IEF166D is N, TNP will not be allowed to start. The issuing of this message is dependent on the z/OS release level being used. It can be avoided by choosing either TNPSTART(STCVS) or TNPSTART(TNPJOB).

If TNPSTART(STCVS) is specified during H210IPZ installation, the JCL that is customized for TNPJOB is used to run TNP as a started task. The job issues the z/OS /*\$VS, ‘S TNPPROC’ command (where TNPPROC is the name specified by the TNPPROC(name) installation parameter, then the batch job terminates.

In all cases, the TNP batch job or started task issues the TNPOFFER, then waits for work requests from remote NetEx/IP Requesters. The TNPOFFER request is only used to receive a notification of NetEx/IP termination.

The TNP batch job to run TNP as a batch program is shown in Figure 28.

The TNP batch job to run TNP as a started task when TNPSTART(STCSTART) is specified is shown in Figure 29.

The TNP batch job to run TNP as a started task when TNPSTART(STCVS) is specified is shown in Figure 30.

The TNP started task PROC is shown in Figure 31.

```
//TNP    JOB  , 'TNP SERVER' ,
//          CLASS=A,MSGCLASS=A
//*
//TNP      EXEC PGM=TNPLISN,MEMLIMIT=4G,REGION=0M
//STEPLIB DD  DISP=SHR,
//          DSN=NETEX.H210IPZ.NTXCLOAD
//*
//*  (OPTIONAL) SYSTCPD POINTS TO ALTERNATE TCP/IP STACK
//*
//SYSTCPD DD  DISP=SHR,
//          DSN=TCPIPT.TCPIP.DATA
//*
//TNPCFG   DD  DISP=SHR,
//          DSN=NETEX.H210IPZ.NTXCTL(TNPCFG)
//TNPLOG   DD  SYSOUT=*,DCB=BUFNO=01
//*
//STDOUT   DD  SYSOUT=*,DCB=BUFNO=01
//STDERR   DD  SYSOUT=*,DCB=BUFNO=01
//SYSPRINT DD  SYSOUT=*,DCB=BUFNO=01
//SYSTEM   DD  SYSOUT=*,DCB=BUFNO=01
//SYSUDUMP DD  SYSOUT=*,DCB=BUFNO=01
//CEEDUMP  DD  SYSOUT=*,DCB=BUFNO=01
//
```

Figure 28. Distributed TNP job used when running TNP as a batch job

```
/*$VS, 'S TNPPROC'
//TNP    JOB  , 'START TNP TASK' ,
//          CLASS=A,MSGCLASS=A,NOTIFY=TNUSER
//*
//TNP      EXEC PGM=IEFBR14
//
```

Figure 29. Distributed TNP job used to start a TNP started task (TNPSTART(STCSTART) specified)

```
//TNP    JOB , 'START TNP TASK',
//          CLASS=A,MSGCLASS=A,NOTIFY=TNPUSER
// *
//TNP    EXEC PGM=IEFBR14
//      START TNPPROC //
```

Figure 30. Distributed TNP job used to start a TNP started task (TNPSTART(STCVS) specified)

```
//TNP      PROC SYSCL=A
//TNP      EXEC PGM=TNPLISN,
//          MEMLIMIT=4G,
//          TIME=1440,
//          REGION=0M
//STEPLIB DD DISP=SHR,
//          DSN=NETEX.NTXCLOAD
// *
// * (OPTIONAL) SYSTCPD POINTS TO ALTERNATE TCP/IP STACK
// *
// *SYSTCPD DD DISP=SHR,
// *SYSTCPD+1 DSN=TCPIP.TCPIP.DATA
// *
//TNPCFG DD DISP=SHR,
//          DSN=NETEX.NTXCTL(TNPCFG)
//TNPLOG DD SYSOUT=&SYSCL,DCB=BUFNO=01
// *
//STDOUT DD SYSOUT=&SYSCL,DCB=BUFNO=01
//STDERR DD SYSOUT=&SYSCL,DCB=BUFNO=01
//SYSPRINT DD SYSOUT=&SYSCL,DCB=BUFNO=01
//SYSTEM DD SYSOUT=&SYSCL,DCB=BUFNO=01
//SYSUDUMP DD SYSOUT=&SYSCL,DCB=BUFNO=01
//CEEDUMP DD SYSOUT=&SYSCL,DCB=BUFNO=01
//
```

Figure 31. Distributed TNP Proc

TNPCFG

Member TNPCFG is the TNP configuration file, and is shown in Figure 32. The PORT keyword specifies the port number used with NetEx/IP Requester hosts, and must be the same as the port specified by the 'TCP' parameter in the NetEx/IP Requester host configuration file. The CMDPORT number specifies a command port used for internal TNP connections. The SSNM parameter specifies the subsystem name of the local NetEx/IP. The DEBUG parameter enables or disables debug tracing to the TNPLOG file. The TNPOP parameter enables or disables the TNP operator command interface. The LOGCMDs parameter enables or disables the echoing of TNP operator commands. The OFFRNM parameter specifies the Netex offer name for TNP. The TRACE parameter specifies the DD name of the TNP message log. The TNPCFG file is built and customized during the H210IPZ installation job. However, this file can also be edited and changed, if needed, anytime after the H210IPZ installation.

```

#
#      TNP CONFIGURATION
#      =====
#
#      KEYWORD      VALUE                                DEFAULT
#      -----
#      TNP          (NONE)
#      PORT          PORT NUMBER USED WITH                5001
#                   NetEx/IP REQUESTER(S)
#      CMDPORT      PORT NUMBER USED FOR                  7002
#                   COMMAND PROCESSING
#      SSNM          LOCAL NETEX SUBSYSTEM NAME           NETX
#      DEBUG         OFF/ON/ON2                           OFF
#      TNPOP         OFF/ON                                ON
#      LOGCMDS       OFF/ON                                ON
#      OFFRNM        NETEX OFFER NAME                     TNPOFFER
#      TRACE         TNP MESSAGE LOG DD                   TNPLOG
#
TNP PORT 5002 CMDPORT 7003 SSNM NETT

```

Figure 32. Requester host configuration file

tnpop

tnpop is a minimal operator interface to dynamically set the TNP configuration parameters which can be set persistantly in the tnp.cfg file. The following commands are supported:

help – display the list of possible tnpop commands

debug (on|on2|off) – set the level of debugging messages to the tnplog (on or off) on2=on

tv – display TNP's internal version

logcmds (on|off) – turning on/off the logging of tnpop commands to the tnplog (and output of valid commands)

servlist – displays any in progress attached tnp server who hasn't started yet (usually there are none)

histlist – displays a running total of TNP session count per requestor

Adding NetEx/IP Requester Hosts to the NCT

For TNP configurations, an additional HOST entry may optionally be defined in the NCT to identify each NetEx/IP Requester host. This NCT change should be made to all of the host-based NetEx/IP NCT's. The name on the HOST statement identifying the NetEx/IP Requester host must be the same name as specified by the 'local' keyword in the NetEx/IP Requester configuration file. Additionally, the following NetEx/IP statement must be added to the NTXINIT initialization file:

```
MULTIH ON
```

The recommended configuration is that the NetEx/IP Requester hosts should be added to the NCT file. Doing this will allow greater control over how connections get established in cases where duplicate NetEx/IP application OFFER names exist on multiple hosts. Without having specific entries in the NCT per Requester

host, and not specifying MULTIH ON, it can be somewhat unpredictable how connections to duplicate OFFER names get established, since all applications would appear as z/OS intrahost connections.

An additional NCT requirement for NetEx/IP Requester hosts is that the ADAPTER defined for the NetEx/IP Requester HOST must be the same as an Adapter defined for the local H210IPZ host, and the Adapter must be identified as an OSA adapter with the OSAA initialization statement.

An example of an H210IPZ NCT with a NetEx/IP Requester host is shown in Figure 33

```
*
** Local HOST ZOST
*
ZOST      HOST      TYPE=IBM      MODEL=Z9
          PROTOCOL=2
*
          OSA Interface
          ADAPTER MODEL=N220  NETADDR=01  CHANADDR=0100
          T0=IP1
          NUMADDRES=4      SMGDREF=00
*
** NetEx/IP Requester host (uses OSA as NetEx/IP intrahost path between
** z/OS NetEx/IP and NetEx/IP Requester host)
*
TNPFLASH HOST      TYPE=IBM      MODEL=Z9
          PROTOCOL=2
*
          OSA Interface
          ADAPTER MODEL=N220  NETADDR=01  CHANADDR=0100
          T0=IP1
          NUMADDRES=4      SMGDREF=00
*
```

Figure 33. Sample NCT with NetEx/IP Requester host

Requester Configuration

The NetEx/IP Requester host also has a configuration file. Refer to the appropriate NetEx/IP Requester manual for a description of the configuration file. It contains the IP address and port number used when communicating with H210IPZ TNP.

The "local" keyword specifies a name for the local client host. If MULTIHOST is set (recommended), this name is used by the TNP NETEX as the NETEX hostname for sessions offered by the local client host. The name specified should be added to the NetEx/IP Network Configuration Table (NCT) on all of the host-based NetEx/IP platforms. (Refer to "Adding NetEx/IP Requester Hosts to the NCT" on page 36).

The "netex" keyword identifies the TNP host. The name following the "netex" keyword identifies a NetEx/IP host that supports the TNP feature. The third value specifies the IP address of the NetEx/IP TNP host. The 'tcp' keyword specifies the port number used by the H210IPZ TNP component, and must be the same as the port specified in the H210IPZ TNP configuration file. If the port number is not specified, the default port number (5001) is used.

An example of a NetEx/IP Requester configuration file is shown in Figure 34.

```
#
# Sample NetEx/IP Requester configuration
#
# The "local" keyword specifies a name for the local client host.
#
local tnpflash
#
# The "netex" keyword identifies the TNP host, along with the IP address and
# port number used by TNP.
#
netex ZOS5T 10.1.5.156 tcp 5001
```

Figure 34. Sample NetEx/IP Requester configuration

Installing NetEx/IP

General

This section provides a detailed step-by-step installation procedure for H210IPZ NetEx/IP. Review the information provided in “NetEx/IP Pre-Installation” on page 13, and complete any pre-installation steps before installing NetEx/IP.

This section discusses the following topics:

- Before Installing H210IPZ NetEx/IP
- Installation Process
- Subsystem Validity Checking

Related Topics

See the following sections for more information:

- “Release Distribution” on page 40 describes the contents of the release distribution.
- “Initialization Statements” on page 73 describes the installation macros.
- “User Exits” on page 149 describes the user exits provided in NetEx.

See the *Configuration Manager and NetEx Alternate Path Retry User Guide* for a description of the Configuration Manager statements.

Release Distribution

H210IPZ NetEx/IP is distributed as a downloadable file.

Distribution Contents

The H210IPZ distribution consists of one file (h210ipz.bin). It is a TSO TRANSMITed file, which when received, results in creating a PDS library that contains two members (NTXINST and NTX0704), which are used to complete the H210IPZ installation. The h210ipz.bin distribution file can either be downloaded from Network Executive Software, or it can be copied from physical media, depending on the obtained distribution. To download the distribution file, contact support at support@netex.com for download instructions.

NTX0704

NTX0704 is a member in the PDS that is created when the H210IPZ.bin distribution file is processed with the TSO RECEIVE command. NTX0704 is another TSO TRANSMITed file that consists of all of the NetEx/IP libraries necessary to complete the installation.

NTXINST

NTXINST is a member in the PDS that is created when the H210IPZ.bin distribution file is processed with the TSO RECEIVE command. NTXINST is a job that extracts the installation libraries from NTX0704, then performs the necessary steps to complete the installation..

LPA Module Considerations

The NetEx/IP distribution contains two modules that reside in the Link Pack Area (LPA):

NXMTERM0

NXMUIF00

The versions of these modules that are distributed with Release 7.4 must be used with NetEx/IP Release 7.4, as prior versions of these modules are incompatible with Release 7.4. Similarly, the Release 7.4 versions of these modules are incompatible with all prior versions of NetEx/IP. If it is desired to allow a Release 7.4 version of NetEx/IP to co-exist with an earlier version of NetEx/IP in the same LPAR, then unique names must be given to the new Release 7.4 LPA modules. For example, if NXMTERM0 and NXMUIF00 already exist in an LPA library for an earlier version of NetEx/IP, then new unique names must be assigned to the Release 7.4 versions. Refer to the following specific installation steps for instructions:

- Step 4. Execute the NTXINST job on z/OS.
- Step 7. Review the H210IPZ initialization parameters.
- Step 10. Activate the NetEx/IP LPA modules.

Before Installing H210IPZ

Before installing H210IPZ NetEx, define the OSA devices to z/OS. Use either the HCD process or the IOCP process, depending on the level supported on the customer's system.

Related Topics

See the following section for more information:

- “z/OS Considerations” on page 15.

Obtaining the Software Key

As part of the installation process, a software key must be obtained from Network Executive Software, Inc. This software key is based on the serial number of the CPU on which NetEx/IP will be used, and on the particular features supported by H210IPZ, and will authorize NetEx/IP to be used on a particular LPAR.

The software key can be obtained in advance by using the following procedure:

1. Issue the following command on the z/OS system on which NetEx/IP will be installed:

```
D M=CPU
```

When this command is issued, it will display the

CPU serial number

LPAR NAME (LP NAME)

LPAR ID (LP ID)

This command should be issued on each LPAR in which H210IPZ will be executed.

An example of the D M=CPU command is shown in Figure 35.

```
D M=CPU
IEE174I 14.27.09 DISPLAY M 981
PROCESSOR STATUS
ID  CPU                      SERIAL
00  +                        01BC7F2096
01  +                        01BC7F2096
02  N

CPC ND = 002096.R07.IBM.02.000000003BC7F
CPC SI = 2096.C02.IBM.02.0000000000003BC7F
CPC ID = 00
CPC NAME = P003BC7F
LP NAME = ZOS1          LP ID = 1
CSS ID = 0
MIF ID = 1

+ ONLINE      - OFFLINE      . DOES NOT EXIST      W WLM-MANAGED
N NOT AVAILABLE

CPC ND  CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC SI  SYSTEM INFORMATION FROM STSI INSTRUCTION
CPC ID  CENTRAL PROCESSING COMPLEX IDENTIFIER
CPC NAME CENTRAL PROCESSING COMPLEX NAME
LP NAME LOGICAL PARTITION NAME
LP ID   LOGICAL PARTITION IDENTIFIER
CSS ID  CHANNEL SUBSYSTEM IDENTIFIER
MIF ID  MULTIPLE IMAGE FACILITY IMAGE IDENTIFIER
```

Figure 35. Output display of 'D M=CPU' command

Note: if keys are needed for a machine on which the ‘D M=CPU’ command cannot be issued (e.g. it is a new machine that is not yet installed, or it is an offsite third-party DR system), you must still provide the same information (machine serial number, model, and LPAR names) in order for the key to get generated.

2. Contact Network Executive Software, Inc. by using either of the following methods:

- telephone at (800) 854-0359
- email: support@netex.com

Please provide the following information:

- Customer name
 - CPU serial number (entire 10 digits)
 - LPAR name(s)
 - NetEx/IP product being installed (H210IPZ)
3. Network Executive Software, Inc. will generate the key(s) and return them by e-mail. This key is required during the installation process.

Installation Process

This section describes the installation procedure for the H210IPZ NetEx/IP Release 7.3 distribution.

The following steps outline the installation process. Before proceeding with the installation, please read the Memo to Users accompanying the distribution for any additions or changes to the installation instructions.

- Step 1. Obtain the H210IPZ distribution file.
- Step 2. Upload the distribution file to z/OS.
- Step 3. TSO RECEIVE the distribution file.
- Step 4. Execute the NTXINST job on z/OS.
- Step 5. Check for required updates.
- Step 6. Obtain the H210IPZ software key.
- Step 7. Review the H210IPZ initialization parameters.
- Step 8. Review the NCT configuration.
- Step 9. Authorize the NTXLOAD load library.
- Step 10. Activate the NetEx/IP LPA modules.
- Step 11. Define OSA devices to VTAM
- Step 12. Define Device and Link statements to TCP/IP
- Step 13. Define RACF OMVS segment for Netex
- Step 14. (Optional) Define RACF OMVS segment for TNP
- Step 15. (Optional) Update Policy Agent
- Step 16. Update DNS
- Step 17. Activate VTAM Nodes
- Step 18. (Optional) Add NETEX entry in Program Properties Table
- Step 19. Review Installed JCL
- Step 20. Start NETEX
- Step 21. (Optional) Execute the NTXMEAT Program
- Step 22. (Optional) Execute the NTXMGGEN Program

Step 1. Obtain the H210IPZ distribution file.

The distribution file 'h210ipz.bin' can be downloaded from Network Executive Software. Contact support@netex.com to obtain the download instructions.

Step 2. Upload the distribution file to z/OS.

FTP (binary mode) the h210ipz.bin file to the z/OS system as follows:

- 1) Connect via FTP to your z/OS system.
- 2) Change the directory to your desired high level qualifier:
`cd 'high-level-qualifier'`
- 3) If necessary, change the location of your local directory to the location of the distribution file:
`lcd 'directory-name'`
- 4) Set the required attributes for the file:
`quote site lrecl=80 blksize=3120 recfm=fb prim=5000 sec=200 blocks`
- 5) Transfer the distribution file in bin mode:
`bin`
`put h210ipz.bin distpkg.xmit`
- 6) Quit your FTP client

Using the above names results in the distribution file residing on z/OS as the following file:

`'high-level-qualifier.DISTPKG.XMIT'`

Step 3. TSO RECEIVE the distribution file.

Issue the TSO PROFILE PROMPT command to be sure prompting is allowed. Then issue the TSO RECEIVE command against the distribution file uploaded in “Step 2. Upload the distribution file to z/OS.” as follows:

`RECEIVE INDSNAME('dsn')`

where 'dsn' specifies the name of the distribution file that was FTP'd to z/OS.

(for example: `RECEIVE INDSNAME('high-level-qualifier.DISTPKG.XMIT')`)

The RECEIVE command will issue the following prompt:

Enter restore parameters or 'DELETE' or 'END'

Reply with:

`DSN('dsn1')`

where 'dsn1' specifies the name of a PDS distribution library that will be created from 'dsn'.

(for example: `DSN('high-level-qualifier.DFILE')`)

The resulting 'high-level-qualifier.DFILE' dataset is a PDS distribution library that contains two members:

`NTXINST`

`NTX0704`

NTXINST is a job that uses NTX0704 to create the H210IPZ libraries and performs the installation.

Step 4. Execute the NTXINST job on z/OS.

The NTXINST installation job consists of four phases:

LOAD	Allocates the H210IPZ base and update installation datasets and loads the datasets from the downloaded distribution.
LINK	Links the H210IPZ load modules.
EDIT	Automatically tailors the H210IPZ NetEx/IP startup proc and TNP startup JCL.
COPY	Copies the tailored NetEx/IP startup proc to the specified PROCLIB library.

Tailor the installation job in NTXINST.

Warning: DO NOT ISSUE “CHANGE xxx ALL” commands against NTXINST. Change the keyword values on an individual basis only.

- a) Change the following to your site requirements:
 - The JOB card
 - The unit name “UNIT=(SYSALLDA,,DEFER)” on the WORK DD card. Change SYSALLDA to a valid unit name matching your site requirements.

- b) Review and tailor the installation parameters contained in NTXINST. At a minimum, the following parameters should be specified:

START(BEGIN)

STOP(END)

TYPE(BASE)

HLQ(hlqname)

DFILE(dfilename)

Note: If a Release 7.4 version is to co-exist in the same LPAR as a prior version, review the requirements of the following parameters and specify as needed:

NTXUIF(NXMUIFxx)

NTXTERM(NXMTERMx)

- c) Submit the NTXINST job. This job will allocate and load the following H210IPZ distribution libraries:

hlq.NTXCTL	Base control library
hlq.NTXMAC	Base macro library (limited distribution)
hlq.NTXSRC	Base source library (limited distribution)

hlq.NTXSAMP	Base sample library
hlq.NTXPARM	Base parameter library
hlq.NTXLKED	Base linkedit statements library
hlq.NTXLOAD	Base load library
hlq.NTXFLOAD	Base api load library
hlq.NTXCLOAD	Base c tools library
hl1.NTXDLOAD	Base distribution load library
hlq.NTXMLPA	Base LPA load library
hlq.PTFCTL	Update control library
hlq.PTFMAC	Update macro library (limited distribution)
hlq.PTFSRC	Update source library (limited distribution)
hlq.PTFOBJ	Update object library
hlq.PTFSAMP	Update sample library
hlq.PTFPARM	Update parameter library
hlq.PTFLKED	Update linkedit statements library
hlq.PTFLOAD	Update distribution load library

NTXINST installation parameters:

HELP Default value: NO
 Allowed values: YES | NO

The HELP parameter is used to produce a description of the H210IPZ z/OS NetEx/IP installation parameters and their usage. The value "YES" will only produce the HELP output, and no other installation job phases will be executed.

START Default value: BEGIN
 Allowed values: BEGIN | LOAD | LINK | EDIT | COPY

The START parameter is used to determine which phase the H210IPZ installation job will be started at. It can be used in conjunction with the STOP parameter to cause only a portion of the installation job to be executed.

STOP Default value: END
 Allowed values: LOAD | LINK | EDIT | COPY | END

The STOP parameter is used to determine which phase the H210IPZ installation job will be stopped at. It can be used in conjunction with the START parameter to cause only a portion of the installation job to be executed. For example, to only execute the LINK phase, code the START and STOP parameters as:

START(LINK)

STOP(LINK)

TYPE

Default value: BASE

Allowed values: BASE | UPDATE

The TYPE parameter is used to provide the type of installation being performed.

BASE indicates a complete product release installation is being performed. New datasets are allocated, and the steps indicated by the START and STOP parameters are performed against those new datasets.

UPDATE indicates a product update is being performed. Datasets previously allocated during the BASE installation are used, and the steps indicated by the START and STOP parameters are performed against those datasets.

Example: TYPE(BASE)

HLQ

Default value: NETEX.NETEX

Allowed values: Any valid data set name qualifier

The HLQ parameter is used to provide the high level qualifier of the data set names used by the installation job. By default, an HLQ of NETEX.NETEX is used, which results in all of the datasets identified in step c) on page 46 being created using names in the following format:

NETEX.NETEX.NTXCTL Base control library

Note: The data sets defined by the HLQ parameter will be deleted and recreated by the LOAD phase when performing a BASE installation.

Example: HLQ(SYSP.NETEX)

COMPRESS

Default value: NO

Allowed values: YES | NO

The COMPRESS parameter is used to determine whether or not the IEBCOPY utility will be invoked to compress the H210IPZ installation data sets before the data set is updated by the installation job. The COMPRESS parameter will only cause a compress of the H210IPZ data sets. Other data sets used by the installation job will not be compressed. System data sets used by the COPY phase will not be compressed prior to the copies.

Example: COMPRESS(NO)

SYSOUT

Default value: *

Allowed values: A-Z, 0-9, *

The SYSOUT parameter is used to provide the JES SYSOUT class for utility output (IEBCOPY, etc.). The value "*" will cause the installation job MSGCLASS SYSOUT class to be used for utility output.

Example: SYSOUT(A)

LINKOUT

Default value: *

Allowed values: A-Z, 0-9, *

The LINKOUT parameter is used to provide the JES SYSOUT class for linkedit output. The value "*" will cause the installation user TSO profile SYSOUT class to be used for linkedit output.

Example: LINKOUT(A)

DFILE

Default value: NETEX.NETEX.DFILE

Allowed values: Any valid fully qualified data set name

The DFILE parameter is used to provide the name of the dataset that was created when the FTP downloaded distribution file was TSO RECEIVE'd in "Step 3. TSO RECEIVE the distribution file."

Example: DFILE(SYSP.PROD.DFILE)

VOLUME

Default value: " (none)

Allowed values: Any valid direct access (DASD) volume name

The VOLUME parameter is used to provide the z/OS volume name which will be used to allocate the H210IPZ installation data sets.

Note: If an SMS policy is in effect for the HLQ name specified, the UNIT and VOLUME parameters can be omitted.

Example: VOLUME(PROD01)

UNIT

Default value: " (none)

Allowed values: Any valid direct access (DASD) unit name

The UNIT parameter is used to provide the z/OS unit name which will be used to allocate the H210IPZ installation data sets.

Note: If an SMS policy is in effect for the HLQ name specified, the UNIT and VOLUME parameters can be omitted.

Example: UNIT(SYSDA)

LINKEDIT

Default value: IEWL

Allowed values: Any valid z/OS program name

The LINKEDIT parameter is used to provide the name of the IBM Linkage Editor or Binder program at your site. This name is usually "IEWL".

If the Linkage Editor or Binder program does not reside in data set SYS1.LINKLIB at your site, the LINKLOAD parameter must be added to the installation job following the LINKEDIT parameter. Its value should be the name of the data set at your site that contains the program specified in the LINKEDIT parameter.

LOADLIB	Default value:	hlq.NTXLOAD (NETEX.NETEX.NTXLOAD)
	Allowed values:	Any valid fully qualified z/OS dataset name
	The LOADLIB parameter is used to provide the name of the load library where the Netex modules will be linked. This library must be an APF authorized library.	
	Example:	LOADLIB(SYSP.PROD.NETEX.LOAD)
CLOADLIB	Default value:	hlq.NTXCLOAD (NETEX.NETEX.NTXCLOAD)
	Allowed values:	Any valid fully qualified z/OS dataset name
	The NTXCLOAD parameter is used to provide the name of the library where the Netex Configuration Manager modules will be linked.	
	Example:	NTXCLOAD(SYSP.PROD.NETEX.NTXCLOAD)
MLPALIB	Default value:	hlq.NTXMLPA (NETEX.NETEX.NTXMLPA)
	Allowed values:	Any valid fully qualified z/OS dataset name
	The MLPALIB parameter is used to provide the name of the library where the Netex LPA modules will be linked. This library must be defined to z/OS as an LPA library.	
	Example:	MLPALIB(SYSP.PROD.NETEX.LPALIB)
FLOADLmm	Default value:	hlq.NTXFLOAD (NETEX.NETEX.NTXFLOAD)
	Allowed values:	Any valid fully qualified z/OS dataset name
	The NTXFLOAD parameter is used to provide the name of the library where the Netex API modules will be linked.	
	Example:	NTXFLOAD(SYSP.PROD.NETEX.NTXFLOAD)
NTXUIF	Default value:	NXMUIF00
	Allowed values:	Any name between 1-8 characters in length

The NTXUIF parameter is used to provide an alternate name for the Netex NXMUIF00 module. This is normally only used if multiple versions of NXMUIF00 must be available for use with different release levels of H210IPZ. NESi will provide guidance when specifying this parameter.

Example: NTXUIF(NXMUIF99)

Note: Release 7.4 adds support that allows NetEx/IP to optionally be run in a reusable address space. The Release 7.4 version of the NXMUIF00 module must be used with NetEx/IP Release 7.4, and is NOT compatible with previous versions. If it is necessary for NetEx/IP Release 7.4 to co-exist with previous versions of NetEx/IP in the same LPAR, then an alternate unique name must be specified for this module. This is done by specifying a different name on the NTXUIF parameter (e.g. NTXUIF(NXMUIF04) when running the NTXINST installation job. This will result in the module being linked into the NTXMLPA library as the name specified. The name specified by the NTXUIF parameter must then also be specified on the UIFNAME statement in the NetEx/IP NTXINIT startup parameters. This will result in the correct version of NXMUIFxx to be loaded by this instance of NetEx/IP. .

NTXTERM Default value: NXMTERM0

Allowed values: Any name between 1-8 characters in length

The NTXTERM parameter is used to provide an alternate name for the Netex NXMTERM0 module. This is normally only used if multiple versions of NXMTERM0 must be available for use with different release levels of H210IPZ. NESi will provide guidance when specifying this parameter.

Example: NTXTERM(NXMTERM9)

Note: Release 7.4 adds support that allows NetEx/IP to optionally be run in a reusable address space. The Release 7.4 version of the NXMTERMx module must be used with NetEx/IP Release 7.4, and is NOT compatible with previous versions. If it is necessary for NetEx/IP Release 7.4 to co-exist with previous versions of NetEx/IP in the same LPAR, then an alternate unique name must be specified for this module. This is done by specifying a different name on the NTXTERM parameter (e.g. NTXTERM(NXMTERMx) when running the NTXINST installation job. This will result in the module being linked into the NTXMLPA library as the name specified. The name specified by the NTXTERM parameter must then also be specified on the TRMNAME statement in the NetEx/IP NTXINIT startup parameters. This will result in the correct version of NXMTERMx to be loaded by this instance of NetEx/IP.

NTXPROC Default value: NETEX

Allowed values: Any valid JCL procedure name

The NTXPROC parameter is used to provide the name for the z/OS started task used to run NetEx/IP. This parameter will also be used as the NTXPROC member name in the dataset defined by the PROCLIB parameter.

Example: NTXPROC(NETEXP)

NTXINIT Default value: hlq.NTXPARM(NTXPARM)
(NETEX.NETEX.NTXPARM(NTXPARM))

Allowed values: Any valid fully qualified z/OS dataset name

The NTXINIT parameter is used to provide the fully qualified name of an existing NetEx/IP NTXINIT initialization file. This name will be added to the NETEX procedure that is built when performing a BASE installation. This parameter is normally used when an existing customer is upgrading from a prior H210IP or H210IPZ release, and would like to incorporate their current NTXINIT initialization file into the new NETEX procedure.

When incorporating a prior NTXINIT file, be sure to review the latest documentation for new or changed NTXINIT parameters.

Example: NTXINIT(SYSP.PROD.NETEX.PARMLIB(NTXPARMP))

NTXPAM

Default value: hlq.PAMFILE
(NETEX.NETEX.PAMFILE)

Allowed values: Any valid fully qualified z/OS dataset name

The NTPAM parameter is used to provide the fully qualified name of an existing NetEx/IP PAM configuration file. This name will be added to the NETEX procedure that is built when performing a BASE installation. This parameter is normally used when an existing customer is upgrading from a prior H210IP or H210IPZ release, and would like to incorporate their current PAM file into the new NETEX procedure.

Example: NTPAM(SYSP.PROD.NETEX.PAMFILEP)

NTXNCT

Default value: hlq.NTXSAMP(SAMPNCT))
(NETEX.NETEX.NTXSAMP(SAMPNCT))

Allowed values: Any valid fully qualified z/OS dataset name

The NTXNCT parameter is used to provide the fully qualified name of an existing NetEx/IP NCT configuration file. This name will be added to the NETEX Configuration Manager utility JCL (CCONFIG, PRTPAM, DUMPPAM) that is built when performing a BASE installation. This parameter is normally used when an existing customer is upgrading from a prior H210IP or H210IPZ release, and would like to incorporate their current NCT file into the new NETEX JCL.

Example: NTXNCT(SYSP.PROD.NETEX.NCTLIB(NCT100))

PRODCONF

Default value: hlq.NTXPARM(PRODCONF)
(NETEX.NETEX.NTXPARM(PRODCONF))

Allowed values: Any valid fully qualified z/OS dataset name

The PRODCONF parameter is used to provide the fully qualified name of an existing NetEx/IP PRODCONF license key file. This name will be added to the NETEX procedure that is built when performing a BASE installation. This parameter is normally used when an existing customer is upgrading from a prior H210IP or H210IPZ release, and would like to incorporate their current PRODCONF file into the new NETEX procedure.

Example: PRODCONF(SYSP.PROD.NETEX.LICENSE(PRODCONF))

LICCODES	<p>Default value: hlq.NTXPARM(LICCODES) (NETEX.NETEX.NTXPARM(LICCODES))</p> <p>Allowed values: Any valid fully qualified z/OS dataset name</p> <p>The LICCODES parameter is used to provide the fully qualified name of an existing NetEx/IP LICCODES license key file. This name will be added to the NETEX procedure that is built when performing a BASE installation. This parameter is normally used when an existing customer is upgrading from a prior H210IP or H210IPZ release, and would like to incorporate their current LICCODES file into the new NETEX procedure.</p> <p>Example: LICCODES(SYSP.PROD.NETEX.LICENSE(LICCODES))</p>
NTXSYSCL	<p>Default value: A</p> <p>Allowed values: A-Z, 0-9</p> <p>The NTXSYSCL parameter is used to provide the JES output class for SYSOUT datasets in the NETEX procedure.</p> <p>Example: NTXSYSCL(H)</p>
NTXRDRCL	<p>Default value: A</p> <p>Allowed values: A-Z, 0-9</p> <p>The NTXRDRCL parameter is used to provide the JES output class for the TNP INTRDR dataset in the NETEX procedure.</p> <p>Example: NTXRDRCL(S)</p>
NTXTRCCL	<p>Default value: A</p> <p>Allowed values: A-Z, 0-9</p> <p>The NTXTRCCL parameter is used to provide the JES output class for the optional SYSTCPT SYSOUT trace dataset in the NETEX procedure.</p> <p>Example: NTXTRCCL(T)</p>
SYSTCPD	<p>Default value: ‘ (none)</p> <p>Allowed values: Any valid fully qualified z/OS dataset name</p> <p>The SYSTCPD parameter is used to provide the fully qualified name of an alternate TCP/IP stack for NetEx/IP usage (OSA only). If this parameter is used, it is specified in the form of an alternate TCPDATA dataset (e.g. TCPIP.TCPIP.DATA). If this parameter is not specified, the default TCP/IP stack will be used by NetEx/IP for OSA connectivity.</p> <p>Example: SYSTCPD(TCPIPZ.TCPIP.DATA)</p>
TNPSTART	<p>Default value: JOB</p>

Allowed values: JOB | STCSTART | STCVS

The TNPSTART parameter is used to specify the default START option for TNP.

If JOB is specified, a customized batch job will be created in 'hlq.NTXCTL', and will be submitted during NetEx/IP initialization to start TNP as a batch job if the TNP option is enabled. The values specified for TNPJOB, TNPJOBCL, TNPACCT, TNPNOTFY, TNPSYSCL, and TNPMSGCL are used to customize the TNP job JCL.

If STCSTART is specified, a customized batch job will be created in 'hlq.NTXCTL', and will be submitted during NetEx/IP initialization to start TNP as a Started Task if the TNP option is enabled. The customized TNP job will issue a z/OS START command (S TNPPROC) to start the TNP PROC that is specified by the TNPPROC parameter. The values specified for TNPJOB, TNPJOBCL, TNPACCT, TNPNOTFY, TNPSYSCL, and TNPMSGCL are used to customize the TNP job JCL that starts the TNP PROC. The values specified for TNPPROC and TNPSYSCL are used to customize the TNP PROC.

NOTE: If the STCSTART option is specified, z/OS message IEFC166D may be displayed when the 'S tnpprocname' command is issued, which requires a Y(es) or N(o) authorization response. If the response to message IEFC166D is Y, TNP will be allowed to continue. If the response to message IEFC166D is N, TNP will not be allowed to start. The issuing of this message is dependent on the z/OS release level being used. It can be avoided by choosing either the STCVS or TNPJOB options.

If STCVS is specified, a customized batch job will be created in 'hlq.NTXCTL', and will be submitted during NetEx/IP initialization to start TNP as a Started Task if the TNP option is enabled. The customized TNP job will issue a z/OS \$VS commands to issue the START command (/*\$VS, 'S TNPPROC') to start the TNP PROC that is specified by the TNPPROC parameter. The values specified for TNPJOB, TNPJOBCL, TNPACCT, TNPNOTFY, TNPSYSCL, and TNPMSGCL are used to customize the TNP job JCL that starts the TNP PROC. The values specified for TNPPROC and TNPSYSCL are used to customize the TNP PROC.

If this parameter is not specified, TNP will be started as a batch job.

Example: TNPSTART(STCSTART)

TNPPROC Default value: TNPPROC

Allowed values: "" or any valid PROC name

The TNPPROC parameter is used to specify the name of the TNP PROC when specifying either STCSTART or STCVS for the TNPSTART parameter. If this parameter is not specified, and STCSTART or STCVS are specified for TNPSTART, the default TNP PROC name of TNPPROC will be used.

Example: TNPPROC(TNPR)

TNPJOB Default value: TNP

Allowed values: ' or any valid job name prefix or TSO user ID.

The TNPJOB parameter is used to specify the job name used on the TNP jobcard when starting TNP as a batch job (TNPSTART(JOB)). It is also used to specify the job name used on

the TNP jobcard when starting TNP as a started task (TNPSTART(STCSTART) or TNPSTART(STCVS)).

Example: TNPJOB(TNPPROD)

TNPJOBCL Default value: A

Allowed values: A-Z, 0-9

The TNPJOBCL parameter is used to specify the job class used on the TNP jobcard when starting TNP as a batch job (TNPSTART(JOB)). It is also used to specify the job name used on the TNP jobcard when starting TNP as a started task (TNPSTART(STCSTART) or TNPSTART(STCVS)).

Example: TNPJOBCL(A)

TNPACCT Default value: '' (none)

Allowed values: " or any valid job accounting information

The TNPACCT parameter is used to specify the job accounting information used on the TNP jobcard when starting TNP as a batch job (TNPSTART(JOB)). It is also used to specify the job accounting information used on the TNP jobcard when starting TNP as a started task (TNPSTART(STCSTART) or TNPSTART(STCVS)).

Example: TNPACCT(1234)

TNPNOTFY Default value: " (none)

Allowed values: " or any valid TSO user ID

The TNPNOTFY parameter is used to specify the TSO user ID NOTIFY parameter used on the TNP jobcard when starting TNP as a batch job (TNPSTART(JOB)). It is also used to specify the NOTIFY parameter used on the TNP jobcard when starting TNP as a started task (TNPSTART(STCSTART) or TNPSTART(STCVS)). To remove the NOTIFY specification from the sample TNP job card, specify this parameter as:

TNPNOTFY("")

Example: TNPNOTFY(USER1)

TNPSYSCL Default value: A

Allowed values: A-Z, 0-9, *

The TNPSYSCL parameter is used to specify the JES output class for SYSOUT datasets in the TNP job when starting TNP as a batch job (TNPSTART(JOB)). It is also used to specify the JES output class for SYSOUT datasets in the TNP job when starting TNP as a started task (TNPSTART(STCSTART) or TNPSTART(STCVS)). The value "*" will cause the TNPMMSGCL value to be used for the SYSOUT datasets in the TNP job.

Example: TNPSYSCL(H)

TNPMSGCL	<p>Default value: A</p> <p>Allowed values: A-Z, 0-9</p> <p>The TNPMSGCL parameter is used to specify the JES output class for the MSGCLASS parameter in the TNP job when starting TNP as a batch job (TNPSTART(JOB)). It is also used to specify the JES output class for the MSGCLASS parameter in the TNP job when starting TNP as a started task (TNPSTART(STCSTART) or TNPSTART(STCVS)).</p> <p>Example: TNPMSGCL(H)</p>
TNPSPORT	<p>Default value: 5001</p> <p>Allowed values: Any valid port allowed by your installation</p> <p>The TNPSPORT parameter is used to provide the port number to be used for TNP TCP/IP connections from NetEx/IP Requester clients.</p> <p>Example: TNPSPORT(6000)</p>
TNPCMDP	<p>Default value: 7002</p> <p>Allowed values: Any valid port allowed by your installation</p> <p>The TNPCMDP parameter is used to provide the port number to be used for TNP command processing.</p> <p>Example: TNPCMDP(7000)</p>
TNPSSNM	<p>Default value: NETX</p> <p>Allowed values: Any 4-character name beginning with NET</p> <p>The TNPSSNM parameter is used to provide the internal subsystem name of NetEx/IP. The name specified must be the same name as specified by the SUBSYS parameter in the NetEx/IP initialization parameters (NTXINIT).</p> <p>Alternate subsystem names are normally only used when running multiple instances of NetEx/IP in the same LPAR.</p> <p>Example: TNPSSNM(NETZ)</p>
PROCLIB	<p>Default value: (None)</p> <p>Allowed values: " or any valid fully qualified data set name</p> <p>The PROCLIB parameter is used to provide the fully qualified name of the JCL procedure library data set that will contain the NetEx/IP startup procedure, as specified by the NTXPROC parameter. The EDIT phase creates the started task JCL procedure and the COPY phase copies it into your specified JCL procedure library for later activation. To bypass the copy of the JCL procedure, either do not execute the COPY phase of the installation job, or specify this parameter as:</p> <p>PROCLIB("")</p>

Example: PROCLIB(SYSP.NETEX.PROCLIB)

REPLACE Default value: NO
Allowed values: YES | NO

The REPLACE parameter is used to determine whether or not existing members in the dataset specified by PROCLIB will be replaced during the COPY phase.

Note: If the NETEX proc name specified by NTXPROC already exists in the PROCLIB dataset specified by PROCLIB, be sure to specify REPLACE(YES).

Example: REPLACE(YES)

DISP Default value: OLD
Allowed values: OLD | SHR

The DISP parameter is used to determine whether the dataset specified by PROCLIB will be allocated for exclusive use during the COPY phase. To ensure being able to successfully copy into the dataset specified by PROCLIB, specify this parameter as:

DISP(SHR)

Step 5. Check for required updates.

Refer to the H210IPZ Memo-to-Users for instructions on checking for product updates at www.netex.com.

Step 6. Obtain the H210IPZ software key.

Obtain the appropriate H210IPZ software key from Netex support (www.netex.com), by providing the output of the 'D M=CPU' command (see "Obtaining the Software Key" on page 42). When the key is received, update the LICCODES file with the new key.

The PRODCONF file consists of a record that contains a LICPATH keyword that identifies the location of the LICCODES file. The LICCODES file contains a record that identifies the actual software key provided by Network Executive Software, Inc. Absence of the correct software key will prevent H210IPZ from executing properly.

A sample PRODCONF file is illustrated in Figure 36. A sample LICCODES file is illustrated in Figure 37.

The LICCODES file may contain multiple keys. If H210IPZ is installed on multiple LPARs on the same system, and if the PRODCONF and LICCODES files are shared across these multiple LPARs, then keys for all of these instances of H210IPZ can be placed in the same LICCODES file. Each instance of H210IPZ will use the first key found that contains its fingerprint.

If there are multiple keys for the same LPAR, H210IPZ will use the first key found for the LPAR on which H210IPZ is being started, as it sequentially reads the keys from the file. This makes it important to add new H210IPZ keys for an existing LPAR to the front of the file. For example, if a new key is installed that provides a license date extension, or adds a new feature to H210IPZ, adding this new key to the file before the old key ensures the new key will be used rather than the old key.

The LICCODES file may also contain keys in the old format (i.e. pre-7.1 keys), which means the same LICCODES file can be shared for both Release 7.3 as well as with prior releases. If NESi's eFT213 product is installed, it can also share the same PRODCONF and LICCODES files with H210IPZ.

The LICCODES files may also contain comments, as shown in Figure 37.

A PRODCONF DD statement must be included in the H210IPZ startup JCL. If the DDN format is used for the LICPATH specification, then a LICCODES DD statement must also be included in the H210IPZ startup JCL, and it must specify the name of the LICCODES file. If the DSN format is used for the LICPATH specification, then the dsname (or dsname/membername) specified by LICPATH identifies the LICCODES file, and a LICCODES DD statement is NOT required in the H210IPZ startup JCL. However, in both cases, the LICCODES file must exist, and be able to be allocated and checked during the execution of H210IPZ.

Sample JCL to load the PRODCONF and LICCODES files is contained in the "hlq.NTXCTL" file, as members LOADLICP (to load PRODCONF) or LOADLIC (to load LICCODES).

```
LICPATH //DDN:LICCODES
-- or --
LICPATH //DSN:dsname
-- or --
LICPATH //DSN:dsname(membername)
```

Figure 36. Sample PRODCONF records

```
*
* LICCODES Software Keys File
* This file may contain comments, indicated by an "*" or "#" in column 1
*
*   NTX210IP IP OSA
* B6YL-6AEA-BAMF-UAH7-AH7Q-CAPU-GAHS-NYX5-5DZE-ZB26
*   NTX210IP OSA TNP
CEPL-6AEA-BADF-UAAM-BQA6-MGAH-TBH3-I4YC
```

Figure 37. Sample LICCODES record

Step 7. Review the H210IPZ initialization parameters.

A sample initialization file is located in hlq.NTXPARM(NTXPARM). Refer to "Initialization Statements" on page 73 for a description of the available statements.

After the installation, member NTXPARM should be in the data set described by the NTXINIT DD statement in the NetEx start-up JCL. Edit member NTXPARM to your site specifications. Change the hostname variable to match the hostname specified in "Step 8. Review the NCT configuration." and save the changes. If OSA device(s) are going to be used in the configuration, identify the OSA devices in the NTXINIT initialization file. Refer to the "OSAA" initialization statement on page 111.

Step 8. Review the NCT configuration.

For this step, use your existing NCT, or use the sample NCT contained in hlq.NTXSAMP(SAMPNCT).

Review the NCT and make any necessary changes. Refer to “OSA NCT Considerations” on page 65 for a description of NCT OSA requirements.

Run the Configuration Manager to create the NetEx/IP configuration file. Refer to the “Configuration Manager” section on page 65 for details on running the Configuration Manager. Sample JCL to run Configuration Manager is located in hlq.NTXCTL(CCONFIG).

Step 9. Authorize the NTXLOAD load library.

Use your site procedures to ensure the hlq.NTXLOAD library is authorized. For example, this can be accomplished by adding it to either the ‘IEAAPFxx’ or ‘PROGxx’ PARMLIB members, depending on the customer method in use for authorizing libraries. If you are using the ‘PROGxx’ method, and you have the dynamic APF list format specified, then the NTXLOAD library can be authorized dynamically without requiring an IPL. If you are using the ‘IEAAPFxx’ method, or the static APF list format is specified in the ‘PROGxx’ member, then an IPL is required to authorize the NTXLOAD library.

Refer to the IBM publication “*z/OS Initialization and Tuning Guide*” for complete descriptions of the methods used to authorize libraries.

Step 10. Activate the NetEx/IP LPA modules.

Activate the NetEx/IP LPA modules NXMTERM0 and NXMUIF00 (or alternate names used). Depending on where you placed these modules during the installation (LPALIB or an MLPALIB), an IPL (with CLPA) or an MLPA refresh will be required. If an MLPALIB was used, be sure to update the appropriate IEALPAXX member to include this library. Alternatively, NXMTERM0 (or alternate name used) may be moved to either the NetEx/IP load library or to an authorized LNKST library.

NXMUIF00 (or alternate name used) must be linked into LPALIB or an MLPALIB library.

NXMUIF00 and NXMTERM0 (or alternate names used) can be dynamically activated into LPA by issuing the following z/OS commands:

```
SETPROG LPA,ADD,MODNAME=NXMUIFxx,DSNAME=name-of-ntxmlpa-library
SETPROG LPA,ADD,MODNAME=NXMTERMx,DSNAME=name-of-ntxmlpa-library
```

Examples: SETPROG LPA,ADD,MODNAME=NXMUIF01,DSNAME=SYSP.NETEX.LOAD
 SETPROG LPA,ADD,MODNAME=NXMTERM1,DSNAME=SYS2.MLPALIB2

However, this activation is only good for the life of the IPL. To make these values persistent, add them to the appropriate IEALPAXX member.

Refer to the “*z/OS Initialization and Tuning Reference Manual*” for descriptions of the LPALSTxx and IEALPAXX members.

Step 11. Define OSA devices to VTAM

This step is required when using OSA devices to provide z/OS NetEx/IP connectivity. OSA devices must be defined to VTAM in a Transport Resource List (TRL) major node. Refer to “VTAM Considerations” on page 19 for a description of what must be done in this step. Sample VTAM TRL definitions are contained in ‘hlq.NTXSAMP’.

Step 12. Define Device and Link statements to TCP/IP

This step is required when using OSA devices to provide z/OS NetEx/IP connectivity; y. Refer to “TCP/IP Considerations” on page 19 for a description of what must be done in this step.

Step 13. Define RACF OMVS segment for Netex

This step is required when using OSA devices to provide z/OS NetEx/IP connectivity. The user name associated with the NETEX started task must have an OMVS segment defined in RACF. Refer to “RACF Considerations” on page 20 for a description of what must be done in this step.

Step 14. (Optional) Define RACF OMVS segment for TNP

This step is required when using the TNP feature. The user name associated with the TNP batch job or started task must have an OMVS segment defined in RACF. Refer to “RACF Considerations” on page 20 for a description of what must be done in this step.

Step 15. (Optional) Update Policy Agent

If z/OS Policy Agent is being used, determine if you require any site required policy changes in order to authorize NetEx/IP access over the IP network.

Step 16. Update DNS

This step is required when using OSA devices to provide z/OS NetEx/IP connectivity. If an OSA-Express interface is being used, the IP host name ‘ntx0000uuss’ identifies the NetEx/IP IP-resolvable host name of the OSA-Express interface; where *uu* and *ss* specify the NETADDR and SMGDREF values respectively, as defined in the NCT by the ADAPTER statement that is identified as an OSA-Express interface by the OSAA initialization statement. The ‘ntx0000uuss’ hostname must resolve to the IP address reserved for NetEx/IP on the corresponding OSA-Express interface. This can be accomplished by updating the site’s Domain Name Server (DNS) with the ‘ntx0000uuss’ hostname and IP address.

As an alternative to using DNS to provide host name IP resolution services, the IP address of a local OSA interface can be specified on the OSAA initialization statement.

Step 17. Activate VTAM Nodes

This step is required when using OSA devices to provide z/OS NetEx/IP connectivity. If the NetEx/IP VTAM node(s) are to be automatically activated during VTAM startup, add the VTAM major node name of

the NetEx/IP OSA-Express TRL to the VTAM configuration list in data set 'SYS1.VTAMLST(ATCCON00), or the equivalent data set in use.

An alternate method to activate the NetEx/IP OSA-Express TRL is to issue the following command as part of the z/OS IPL procedure, after VTAM initialization has completed, or prior to starting NetEx/IP:

```
V NET,ACT,ID=xxxxxxxx
```

where 'xxxxxxxx' specifies the name of the NetEx/IP OSA-Express TRL major node.

Step 18. (Optional) Add NETEX entry in Program Properties Table

If ALLOWUSERKEYCSA is specified (or defaulted) as NO in a DIAGxx PARMLIB member, an entry for NETEX must be added to the Program Properties Table contained in the current SCHEDxx PARMLIB member. Refer to "ALLOWUSERKEYCSA Support" on page 15 for a description of what must be done in this step.

Step 19. Review Installed JCL

Review the NETEX started task JCL, along with other JCL in '&hlq.NTXCTL', and make any necessary changes to satisfy your site requirements.

Step 20. Start NETEX

Start NETEX by issuing the 'S NETEX' command.

Since NetEx/IP is a subsystem, and therefore functions as a long running task, Network Executive Software advises running NetEx as a started task as opposed to a batch job. If this is not possible, then you should give the job enough time on the JOB card so that a time-out condition does not occur.

NetEx establishes a cross-memory environment and z/OS terminates the address space at task termination. This results in the initiator being drained if NetEx is executing as a batch job. When running in an ordinary address space, the ASID in which NetEx was executing is not re-used until after the next IPL. This is a consideration when specifying a value for the z/OS &MAXUSER parameter.

Start NetEx by entering the following command:

```
S NETEX
```

Optionally start NetEx with the REUSASID parameter by entering the following command:

```
S NETEX,REUSASID=YES
```

See "Emergency Override Procedure" on page 63 if NetEx does not initialize.

Note: The test programs in the "hlq.NTXCTL" data set, may not be compatible with other distributions. Only run these as directed by Netex Support.

Step 21. (Optional) Execute the NTXMEAT Program

The NTXMEAT program “offers” its services. This program works in tandem with NTXMGGEN.

Execute the NTXMEAT program. Edit member NTXMEAT, shown in Figure 38 on page 62, in the “hlq.NTXCTL” data set, and change the parameters to your specifications. The “offer name” for both the NTXMEAT and NTXMGGEN jobs must be the same. Save the changes and submit the job to z/OS for execution. This program remains running and must be cancelled when you complete processing.

```
//NTXMEAT JOB CLASS=A,MSGCLASS=A,NOTIFY=&SYSUID,REGION=0M
//*
//*      NOTE: SUBSTITUTE THE APPROPRIATE VALUES IN THE
//*      FOLLOWING JCL
//*
//*      SYSIN PARM: #SESSIONS VALIDATE
//*      #Sessions Validate OffrName HostName SSNM
//*
//CPROG EXEC PGM=NTXMEAT
//STEPLIB DD DISP=SHR,DSN=BETATST.H210IPZI.NTXCLOAD
//SYSPRINT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//STDIN DD *
1 Y NTXMEAT Host NETX
/*
//
//
```

Figure 38. Sample NTXMEAT Job, Member ‘NTXMEAT’ in hlq.NTXCTL

Step 22. (Optional) Execute the NTXMGGEN Program

The NTXMGGEN “connects” to NTXMEAT and sends a variable number (and size) of data blocks to the NTXMEAT program.

Execute the NTXMGGEN program. Edit member NTXMGGEN, shown in Figure 39, in the “hlq.NTXCTL” data set, and change the parameters to your specifications. Specify the value of the host name on which the NTXMEAT program is executing in the PARM field. The “offer name” for both the NTXMEAT and NTXMGGEN jobs must be the same. The values specified in the PARM field of the NTXMGGEN are shown in Figure 39.. Save the changes and submit the job to z/OS for execution. This program ends when the request is complete.

```
//NTXMGEN  JOB CLASS=A,MSGCLASS=A,NOTIFY=&SYSUID,REGION=0M
//*
//*      NOTE: SUBSTITUTE THE APPROPRIATE VALUES IN THE
//*      FOLLOWING JCL
//*
//*  SYSIN PARMS:
//*      #SESSIONS BLOCKS SIZE ODATA LOOPS DMODE VALIDATE HOST OFR SSID
//*
//CPROG      EXEC PGM=NTXMGEN
//STEPLIB    DD  DISP=SHR,
//           DSN=BETATST.H210IPZI.NTXCLOAD
//SYSPRINT   DD  SYSOUT=*
//STDOUT     DD  SYSOUT=*
//STDERR     DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//STDIN      DD  * ,
1 1000 1000 10 1 0 Y NTXLCL01 NTXMEAT NETX
//
//
```

Figure 39. Sample NTXMGEN Job, Member ‘NTXMGEN’ in hlq.NTXCTL

Subsystem Validity Checking

During initialization, NetEx validates the subsystem name as defined in the NTXINIT initialization statements. If it determines that the subsystem name is not valid, NetEx issues an appropriate error message and NetEx terminates. The following provides possible reasons for failing this validity check:

- The specified NetEx subsystem name is already defined in the system as an “active” subsystem.

After initialization, the value of the subsystem name becomes “reserved.” Any attempt to start another NetEx with the same subsystem name at this point will fail the validity checking.

During termination, NetEx “resets” the environment so that NetEx can initialize properly the next time it is started.

It is possible for this value to not get reset properly (for example, if a NetEx ABEND occurs during termination). If this happens, the system remains in a state that prevents NetEx from initializing until after an IPL is issued. In this case (and only in this case), use the emergency override procedure. See “Emergency Override Procedure” for more information.

Emergency Override Procedure

If the system remains in the failed state identified in “Subsystem Validity Checking”, then the emergency override procedure signals NetEx to bypass the subsystem name validity checking during initialization. The emergency override procedure tells NetEx to bypass the subsystem name validity checking by specifying the PARM=FORCE in the start-up JCL. The following example shows the format of the command used when NetEx requires an emergency override:

```
S NETEX,PARM=FORCE
```

CAUTION
Use the emergency override only when a previous NetEx execution failed to complete termination processing. Any other use of this procedure causes unpredictable results in the system.

Direct questions or concerns regarding this procedure to Network Executive Software, Inc.

Configuration Manager

General

The Configuration Manager program reads statements contained in a Network Configuration Table (NCT) that describe the entire network and produces an output file known as the PAMFILE. NetEx uses the physical addressing information from this PAMFILE to access a remote host. Refer to the “Operation” section in the *Configuration Manager and NetEx Alternate Path Retry User Guide* for descriptions of the configuration statements.

NOTE: The Configuration Manager processes all columns of a control statement. Ensure that each statement begins in column 1 and there are no sequence numbers in columns 73-80.

OSA NCT Considerations

No changes were made to either the Network Configuration Table (NCT) or the Configuration Manager to support OSA for NetEx/IP. The mechanism for adapting the NCT to support OSA and IP networks is as follows:

- Use the TRUNK statement to define the NetEx/IP network (a unique name that is used when identifying all NetEx/IP nodes on the NetEx/IP network).
- Use the T0 parameter on the ADAPTER statement(s) of the NetEx/IP host to indicate connectivity on the NetEx/IP network.
- Use the ADAPTER statement to define the OSA device.
- Specify the MODEL type for the OSA device as N220.
- Use the CHANADDR and NUMADDR parameters on the OSA ADAPTER definition to define the OSA devices. In the case of an OSA definition, CHANADDR is an arbitrary number that is only used to satisfy Configuration Manager requirements. It does not need to be a defined device.
- Use the NETADDR and SMGDREF parameters on each NetEx/IP ADAPTER statement to identify a unique NetEx/IP network address.

As part of the installation, a sample NCT containing OSA adapter definitions is located in dataset 'hlq.NTXSAMP(SAMPNCT)'

Sample NCT Definitions

OSA interfaces are defined in the NetEx/IP Network Configuration Table (NCT) in a similar manner as how N other NetEx/IP hosts are defined. For example, a Linux NetEx/IP host could be defined in the NCT as shown in Figure 40.

```
Linux1  HOST  TYPE=LINUX  Model=Fedora3
        PROTOCOL=2
* Linux NetEx/IP at NetEx/IP network address A164
        ADAPTER MODEL=N130  NETADDR=A1
        T0=ETH1  SMGDREF=64
```

Figure 40. NCT definition for Linux NetEx/IP host

An OSA adapter could be defined in the NCT as shown in Figure 41.

```
ZOS2  HOST  TYPE=IBM  Model=Z9
        PROTOCOL=2
* z/OS NetEx/IP at NetEx/IP network address 0300
        ADAPTER MODEL=N220  NETADDR=03  CHANADDR=0300
        T0=ETH1  NUMADDRES=2  SMGDREF=00
```

Figure 41. NCT definition for z/OS NetEx/IP OSA

As with all other NetEx/IP devices, the combination of the NETADDR and the SMGDREF identifies the NetEx/IP Network Address for this device. In this example, 03 is the Network Address (NETADDR) of this OSA interface, and is used by all NetEx/IP hosts on the network. The CHANADDR and NUMADDRES parameters are only required to satisfy the requirements of the NetEx/IP Configuration Manager, and have no relationship to any device definitions. CHANADDR does not even need to be an existing device. It is just an arbitrary 4-digit hexadecimal number, but it must be unique within the NCT definitions. The SMGDREF value must be within the range of CHANADDR and CHANADDR plus (NUMADDRES-1).

As with all other NetEx/IP devices, the Tx parameter identifies a NetEx/IP connection. In the above examples, a NetEx/IP path exists between Linux1 and ZOS1; Linux1 and ZOS2; and ZOS1 and ZOS2 since they are all identified as being on a common ETH1 network.

In addition to adding the OSA interface to the NetEx/IP NCT, the definition shown in Figure 42 must also be added to the NetEx/IP defaults file (identified by the NTXINIT DD statement in the NetEx/IP startup PROC). This statement directs NetEx/IP to use socket interface calls to the TCP/IP stack for adapter NETADDR 03, instead of using channel I/O to adapter NETADDR 03.

```
*** the following initialization statement identifies
*** NCT adapter "NETADDR 03 SMGDREF 00" as an OSA Adapter
OSAA 0300
```

Figure 42. OSAA definition in NTXINIT initialization file

IP Name Resolution

As in all versions of NetEx/IP, NetEx IP-host names (ntx0000uuss) are used for resolution to IP addresses. In the NCT example for host Linux1, an attempt will be made to resolve host name NTX0000A164 to an IP address. The host name is made up of the NETADDR and SMGDREF values. In this example, this IP address represents the IP address of a network interface on the Linux host.

In the example for host ZOS2, an attempt will be made to resolve host name NTX00000300 to an IP address. In this example, this IP address represents the IP address of an Ethernet port on an OSA interface.

Configuration Manager Control Statements

This section describes the following control statements used when running the Configuration Manager software. They are described starting on page 69.

- DESELECT
- EXIT
- MAKEPAM
- SELECT

The following steps describe how to use the Configuration Manager.

1. Member CONFSAMP (located in “hlq.NTXSAMP”) is a sample Network Configuration Table (NCT). Use it as a guide to create a text file that describes your NetEx/IP configuration.
2. Member CCONFIG (located in “hlq.NTXCTL”) is sample JCL to execute the Configuration Manager (shown in Figure 43). Modify the JCL to meet your requirements. Be sure to change CONFFILE to point to the NCT created in step 1. Change the ‘hostname’ variable in the STDIN DD data set to indicate the Netex host name for which the Configuration Manager is being run.
3. Run the Configuration Manager. The Configuration Manager reads the command statements from the STDIN DD data set. Configuration Manager statements are defined beginning on page 69. Submit the modified CCONFIG job to z/OS for execution. You should receive a return code of 0. If you receive errors as a result of this step, correct the errors in the Network Configuration Table and rerun the job. Explanations for Configuration Manager errors are provided in the “Understanding Configuration Manager Messages” in the *H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages and Abend Codes*. Refer to the “NCT Configuration File Errors” section in the *Configuration Manager and NetEx Alternate Path Retry User Guide* for additional information.

Figure 43 is an example of the CCONFIG job for the Configuration Manager.

```

//CCONFIG      JOB,                Accounting information;
//              CCONFIG,           Process name;
//              CLASS=A,           Standard execute class;
//              MSGCLASS=A,NOTIFY=&SYSUID
//*
//*          SAMPLE JCL to execute the "C" Configuration Manager.
//*
//*          Note: The "C" Configuration Manager does not require
//*          a pre-initialized PAM file.
//*
//*          Note: Substitute the appropriate values in the following
//*          JCL DD-statements for:
//*          "disk" "volser"
//*
//CCONFIG EXEC PGM=CM,                "C" Configuration Manager;
//              PARM='-N, //DDN:CONFFILE', specify NCT DDNAME;
//              REGION=4096K          specify region
//STEPLIB DD DISP=SHR,
//              DSN=NETEX.NTXCLOAD
//*
//*          Configuration Manager commands go here.
//*
//*          The "C" Configuration Manager requires that the
//*          DDNAME of the PAM dataset be specified by a
//*          preceding "DD:" to identify it as a DDNAME.
//*
//*          The command statements MUST start in column 1.
//*
//*          Substitute "hostname" with the Netex hostname
//*          for which the PAM is being created.
//*
//STDIN DD *,DCB=BLKSIZE=080          Configuration Manager commands
SELECT *
MAKEPAM hostname //DDN:OUTFILE
EXIT
//*
//*          Output PAM file definition.
//*
//OUTFILE DD DISP=(,CATLG,DELETE),
//              DSN=NETEX.PAMFILE,
//              SPACE=(0256,(005,010),RLSE),
//*              UNIT=disk,           if needed
//*              VOL=SER=volser,      if needed
//              DCB=(RECFM=F,BLKSIZE=0256,LRECL=0256)
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//SYSTEM DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//*
//*          Network configuration table goes here.
//*
//CONFFILE DD DISP=SHR,
//              DSN=iNETEX.NTXSAMP(SAMPNCT)
//

```

Figure 43. Sample Configuration Manager Job Member “CCONFIG” in hlq.NTXCTL

DESELECT

The Deselect command causes specified host and group names not to be placed in the resulting Physical Address Map (PAM) file. You can specify the host or group names with one or more Deselect commands. This command allows you to remove any host or group name implicitly or explicitly defined by the SELECT command.

The Deselect command has the following format:

Command	Operand
DESELECT	<i>HOSTNAME HOSTNAME . . .</i>

HOSTNAME

Specifies a NetEx host name or group name defined in the configuration file, or NTXLCL.

Related Topics

See the following sections for more information:

- “SELECT” on page 71

EXIT

The EXIT command stops the Configuration Manager.

The EXIT command has the following format:

Command
EXIT

MAKEPAM

The MAKEPAM command creates a PAM file for a specified host and writes it to the defined output file DDNAME. If there are no preceding SELECT or DESELECT statements, then all hosts on the network are selected.

Generally, issuing the MAKEPAM command produces a PAM for the local host. However, you can use the MAKEPAM command to produce a PAM file for any host on the network. Do this for any operating systems that do not have a Configuration Manager (for example, TPF). The PAM files produced by this command are loaded for the remote NetExes. If you use this command to produce a PAM file for multiple hosts, remember to place each PAM produced in separate OUTPUT file DDNAMES.

The MAKEPAM command has following the format:

Command	Required Operand	Optional Operand (Select One)
MAKEPAM	<i>HOSTNAME OUTPUT-FILE-DDNAME</i>	<u>CONNAPR</u> FULLAPR

HOSTNAME

Specifies the name of the host for which the PAM file is being created. This can be the local host. The local host uses the binary file produced by the MAKEPAM command. All paths are displayed. The output of this step is placed in a file called output-file-ddname in “hlq.PAMFILE”. Member CCONFIG (which was created in “Step 4. Execute the NTXINST job on z/OS.”) contains an example of the JCL required for the Configuration Manager.

OUTPUT-FILE-DDNAME

Specifies the DDNAME of the DD JCL statement that references the PAMFILE data set. Refer to the STDIN DD statement in Figure 43 on page 68 for an example.

CONNAPR

Uses the largest possible segment size for APR on each path for every established connection. NetEx will not change the segments sizes on each path. CONNAPR is the default.

FULLAPR

Adjusts segment sizes for APR on each path for every established connection to the smallest size possible. APR can not switch from one path with specified segment size to a path with a smaller segment size. FULLAPR reduces the segment size on all paths to the lowest segments size to ensure APR on all paths.

Related Topics

See the following sections for more information:

- “DESELECT” on page 69
- “SELECT” on page 71

SELECT

The SELECT command selects which hosts will be included as destinations in the resulting PAM file. Omitting the SELECT statement or specifying SELECT * selects all hosts.

By issuing the SELECT command with one or more host names, you are limiting the number of hosts to be used as destinations. You can specify these host names with one or more SELECT commands. You may also specify a group name in the SELECT statement as a HOSTNAME.

The SELECT command has the following format:

Command	Operand
SELECT	<i>HOSTNAME</i> . . . <i>HOSTNAME</i> *

HOSTNAME

Specifies the NetEx host name (identified in the configuration statements). Specify the HOSTNAME for each defined NetEx.

*

Selects all hosts.

Related Topics

See the following sections for more information:

- “DESELECT” on page 69

Initialization Statements

General

This section describes NetEx initialization statements, which define NetEx default values.

Initialization Statements

The NetEx installation system programmer uses the initialization statements to define installation parameters to NetEx. All initialization statements not specified by the installation system programmer use the defaults provided by Network Executive Software.

The HOST NAME statement is the only required statement. The other initialization statements are optional based on your site's specific needs. Most initialization statements can also be temporarily entered as parameters specified on SET operator commands. See the *H210IPZ NetEx/IP for z/OS Operating Systems Operator Reference Manual* for information on the SET command.

Initialization statements are contained in the file referenced by the NTXINIT DD statement. Either an equal sign or one or more spaces can be used as a delimiter between keywords and values.

Table 4 summarizes the initialization statements in alphabetical order, as they are presented in this section.

NOTE: When building an initialization file, place comments on a separate line from the initialization statements. If a comment is on the same line as the initialization statement, NetEx processes it as an additional initialization parameter, and generates an error message.

Table 4. Initialization Statements			
Statement	Default	Recommended	Description
ALIAS	NONE		Alias host names
BACKOFF	50		I/O error retry delay
BITMINT	3600		Timer NIT mask clear
CFOTM	30		Configuration receiver offer time
CMDNO	20		Number of command buffers
CMDPRE	/		Command prefix for NetEx operator commands
CONFIG	NO	YES	Selects Configuration Manager use
CONTIME	30		Time allowed for connect to generate response
DDMAXMLT	6		IDLEMAX multiplier for inactive dead time
DEADTIME	60		Time before silent remote host assumed dead
DEFBI	4096	32K	Default incoming block size
DEFBO	4096	32K	Default outgoing block size
DESCCD	5		Default WTO descriptor code
DFRT	0	0	Default rate
DISE	750(75.0%)		Delay increment send equivalent
DIRE	750(75.0%)		Delay increment receive equivalent
DOWD	0		Default one-way delay of network path, in milliseconds

Table 4. Initialization Statements

Statement	Default	Recommended	Description
DQHB	20000000	20000000	High watermark threshold of data queue
DQLB	10000000	10000000	Low watermark threshold of data queue
DRIVAVG	2		Average number of Driver sessions
DRIVLIM	8		Upper limit of DRIVMAX
DRIVMAX	8		Number of concurrent NetEx Driver sessions allowed
ENDINI1			End initialization processing
FMSML	OFF	OFF	Force multiline WTO rmtoper disp
FUBS	OFF	OFF	Free unused buffer space
GTRACE	OFF		Enables system internal trace for NetEx.
HOST NAME	OS390		Host name
IDLEMAX	240		Maximum idle time when using inactive scheme
IDLETIME	5		Time before an idle message is sent
INACTIVE	OFF		Inactive idle time
IOCOUNT	OFF		Initializes I/O tracking
LARNAME	NXMLAR		Linkage-assist module name
LOGCMD	ON		Log NetEx/IP operator commands
MAXBI	61440	61440	Maximum incoming block size
MAXBO	61440	61440	Maximum outgoing block size
MAXDDBQ	6		Maximum number of queued DDBs
MAXDREAD	2		Maximum number of DREADs
MAXDWRT	2		Maximum number of DWRITES
MAXODATA	3072		Maximum outgoing ODATA block size
MAXSEG	32767	64000	Maximum segment size for this host
MBFI	5		Multiple buffer count inbound
MBFO	5		Multiple buffer count outbound
MFCP	0	0	Memory free copy ratio
MIHTM	15		Missing interrupt handler interval
MKBS	0		Maximum rate of network path, in kilobits per second
MSGLVL	8		Minimum message level displayed
MSGML	ON		Multiple lines for WTO messages
MSGTM	OFF		Timestamp on operator messages
MSGXL	OFF		Operator messages translated into uppercase
MULTIH	OFF	ON when using TNP	Requester HOST name used in connections
NTXOPER	OFF	NTXOPER=ON CLASS=G	Remote operator permitted
OSAA	UUSS		Identifies local adapter as OSA
OSA1IF	ON	OFF	ON = Bind to IP address 0
OUTCLASS	A		Output from NetEx log function
PAGENIT	0		Extra 4k-pages for NITs beyond what NetEx normally assigns
PAGESAVE	100		Extra pages of subroutine save areas beyond what NetEx normally assigns

Table 4. Initialization Statements			
Statement	Default	Recommended	Description
PAGEXTRA	4		Number of extra pages for overflow of any memory size
PAGE16	0		Extra 4K-pages of memory size 16 bytes beyond what NetEx normally assigns
PAGE32	0		Extra 4K-pages of memory size 32 bytes beyond what NetEx normally assigns
PAGE64	0		Extra 4K-pages of memory size 64 bytes beyond what NetEx normally assigns
PAGE128	50		Extra 4K-pages of memory size 128 bytes beyond what NetEx normally assigns
PAGE256	0		Extra 4K-pages of memory size 256 bytes beyond what NetEx normally assigns
PAGE512	0		Extra 4K-pages of memory size 512 bytes beyond what NetEx normally assigns
PAGE1K	2000		Extra 4K-pages of memory size 1K bytes beyond what NetEx normally assigns
PAGE4K	0		Extra 4K-pages of memory size 4K bytes beyond what NetEx normally assigns
PAGE8K	0		Extra 4K-pages of memory size 8K bytes beyond what NetEx normally assigns
PAGE16K	0		Extra 4K-pages of memory size 16K bytes beyond what NetEx normally assigns
PDQNAME	NXMSRBR1		SRB PURGEDQ module name xxxxxxxxx value
PRFP	2		Preferred protocol
RDDP	250(25.0%)		Rate delay decrement % (prot4)
RDII	60		Round trip delay increase intvl
READTIME	60	300	Time NetEx retains unread data
ROUTCD	2		WTO route code
RQPH	855(85.5%)		Receive rate % high (prot4)
RQPL	500(50.0%)		Receive rate % low (prot4)
RQPS	750(75.0%)		Receive rate % equivalence (prot4)
RRCI	2		Receive rate calculation intvl
SESAVG	32		Average number of sessions
SESLIM	32		Upper limit of SESMAX
SESMAX	32		Number of concurrent NetEx sessions allowed
SMFTYPE	00		Establishes the SMF record identifier
SMWTM	15		Session manager read timeout
SNDGRNM	ON		Send group name on connects

Table 4. Initialization Statements

Statement	Default	Recommended	Description
SRBNAME	NXMSRB01		SRB name
SRCI	4		Send rate calculation intvl
SRPH	500(50.0%)		Send rate % high
SRPL	100(10.0%)		Send rate % low
SRPS	200(20.0%)		Send rate % equivalence
SRUI	20		Send rate update interval
SSINAME	NXMSSI		Subsystem interface name
STRP	750(75.0%)		Starting rate %
SUBSYS	NETX		NetEx subsystem name
SWTNAME	NXMSWTPC		SWAIT PC module name
TRACE	OFF		Trace options
TRCNO	4		Number of trace buffers
TRCSIZE	8176		Trace buffer size
TRMNAME	NXMTERM0*		Termination Resource Manager module name
TWOPART	0		Two part message interval
UDPP	6950		UDP port used by NetEx/IP
UIFNAME	NXMUIF00*		User interface name
USERNAME	NXMUSRPC		User program-call module
WDOGINT	2		Time between checking and updating timer values
XCONF	OFF		Confirm exit
XCONN	OFF		Connect exit
XCXN	OFF		Connection exit
XDATAI	OFF		Data input exit
XDATAO	OFF		Data output exit
XDISC	OFF		Disconnection exit
XINIT	OFF		Initialization exit
XOFFR	OFF		Offer exit
XTERM	OFF		Termination exit

- * Note: Release 7.4 adds support that allows NetEx/IP to run in a reusable address space. If Release 7.4 is to co-exist in the same LPAR with previous versions, alternate unique names must be specified for the NTXUIF and NTXTERM installation parameters, as well as the TRMNAME and UIFNAME initialization statements. Refer to the following specific installation steps for instructions:
- Step 4. Execute the NTXINST job on z/OS.
- Step 7. Review the H210IPZ initialization parameters.
- Step 10. Activate the NetEx/IP LPA modules.

ALIAS

The ALIAS statement specifies an alternative name for a configured remote host. This statement may be repeated as necessary.

The ALIAS statement has the format:

Keyword	Required Parameters
ALIAS=	AKA= <i>ALIASNAME</i> REAL= <i>HOSTNAME</i>

AKA

Specifies the alias name of the actual configured host name.

ALIASNAME

Specifies the alternative name for the hostname.

REAL

Specifies the actual name of the configured host.

HOSTNAME

Specifies the actual configured host name.

Example

```
ALIAS= AKA = TWAIN REAL = CLEMENS
```

Related Topics

See the following operator commands in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual:

- “DISPLAY ALIAS”
- “ALIAS”
- “DEALIAS”

BACKOFF

The BACKOFF statement defines the initial delay in milliseconds between 1080 and 10C0 I/O error retries. Each retry delay is calculated by multiplying the retry number by this value.

If this statement is not coded, the default is BACKOFF=50 milliseconds.

The BACKOFF statement has the format:

Keyword (Select One)
BACKOFF= <i>DELAY-TIME</i>
BACKOFF=50

DELAY-TIME

Specifies the delay in milliseconds (0-2000).

Example

```
BACKOFF=60
```

BITMINT

The BITMINT statement defines the time interval between resetting of the output subchannel mask in the NetEx DEVBLOCKs.

If this statement is not coded, the default is BITMINT=3600 seconds.

The BITMINT statement has the format:

Keyword (Select One)
BITMINT= <i>TIMER-INTERVAL-VALUE</i>
BITMINT=3600

TIMER-INTERVAL-VALUE

Specifies the number of seconds (0-9999) before the bit masks are cleared.

Example

```
BITMINT=5400
```

CFOTM

The CFOTM statement specifies the amount of time (in seconds) that the configuration receiver OFFERs, waiting for the configuration sender to CONNECT.

The CFOTM statement has the format:

Keyword (Select One)
CFOTM= <i>SECONDS</i>
CFOTM=30

SECONDS

Specifies the time (in seconds) the configuration receiver (NTXPAMRC) offers before a timeout. Allowed values are 1-999; the default is 30 seconds.

Example

```
CFOTM=60
```

Related Topics

- The “SET CFOTM” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

CMDNO

The CMDNO statement defines the number of command buffers used for in-process operator commands. These buffers store operator commands until they are accepted by NetEx.

The CMDNO statement has the format:

Keyword (Select One)
CMDNO= <i>NUMBER-OF-BUFFERS</i>
CMDNO=20

NUMBER-OF-BUFFERS

Specifies the decimal number of command buffers assigned to NetEx. If this statement is not coded, the default is CMDNO=20.

Example

```
CMDNO=22
```

CMDPRE

The CMDPRE statement defines a one-character command prefix used as the first character of any NetEx operator command.

IMPORTANT: Do not specify a command prefix that is currently being used by another subsystem.

If this statement is not coded, CMDPRE=/ is the default. Do not specify the default if you want to use “/”.

The CMDPRE statement has the format:

Keyword
CMDPRE= <i>PREFIX</i>

PREFIX

Specifies a one-character prefix used on all NetEx operator commands. The default is /.

NOTE: Do not use the following characters as a command prefix because the NetEx scan routine cannot parse them.

- (left parenthesis
-) right parenthesis
- = equal sign
- \ back slash
- | broken vertical bar
- ' single quote
- “ double quote
- [left bracket
-] right bracket
- , comma
- / slash
- ‘ accent
- ~ tilde

Example

CMDPRE=*

CONFIG

The CONFIG statement specifies whether to use the Configuration Manager or the NTCROUTE statements. The Configuration Manager is described in “Installing NetEx” on page 39 and the Configuration Manager and NetEx Alternate Path Retry (APR) User Guide.

WARNING: Support for defining the NetEx/IP configuration with NTCROUTE macros is expected to be dropped in a future release. Network Executive Software recommends using the Configuration Manager to define the NetEx/IP configuration.

Please notify support@netex.com before using the NTCROUTE macros.

The CONFIG statement has the format:

Keyword (Select One)
CONFIG=YES
CONFIG=NO

YES

Enables configuration processing (and may optionally be specified using CFLG=Y in NTCROOTG). Network Executive Software recommends specifying CONFIG=YES.

NO

Disables configuration processing. NO is the default (and may optionally be specified using CFLG=N in NTCROOTG).

Example

```
CONFIG=YES
```

Related Topics

- See “Installing NetEx/IP” on page 39.
- See the *Configuration Manager and NetEx Alternate Path Retry (APR) User Guide*.

CONTIME

The CONTIME statement specifies the maximum number of seconds that NetEx waits for a transport connect message to generate any response from the remote host. If this time is exceeded, the transport protocol assumes that the remote host or remote NetEx is down and returns a Disconnect Indication to the transport caller. The session retries another connection on an alternate path if one is available. Note that CONTIME divided by IDLETIME determines the number of physical I/O attempts to establish a connection.

The CONTIME statement has the format:

Keyword (Select One)
CONTIME= <i>CONNECT-TIMEOUT</i>
CONTIME=30

CONNECT-TIMEOUT

Specifies the number of seconds (0-999) before timeout. If this statement is not coded, CONTIME=30 seconds is the default.

Example

```
CONTIME=60
```

Related Topics

- The “SET CONTIME” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DDMAXMLT

The DDMAXMLT statement specifies the number of idle messages that may be lost before the session gives up and disconnects. This statement depends on the inactive idle scheme (INACTIVE=ON).

The DDMAXMLT statement has the format:

Keyword (Select One)
DDMAXMLT= <i>MULTIPLIER</i>
DDMAXMLT=6

MULTIPLIER

Specifies the number of IDLE messages (1-99) that may be lost before timing out. If this statement is not coded, the default is DDMAXMLT=6.

Example

```
DDMAXMLT=10
```

Related Topics

- The “SET DDMAXMLT” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DEADTIME

The DEADTIME statement specifies the number of seconds that transport software waits until it assumes a disconnection because the remote host does not communicate. The local host transport software generates an “idle” message every IDLETIME seconds; the remote NetEx transport software, if operational, generates an “idle” message of its own. If any message is received from the remote host, DEADTIME is reset. When DEADTIME expires, the transport connection terminates.

NOTE: The session is considered interrupted if a period of time greater than three times IDLETIME is exceeded.

The DEADTIME statement has the format:

Keyword (Select One)
DEADTIME= <i>DEADMAN-TIMEOUT</i>
DEADTIME=60

DEADMAN-TIMEOUT

Specifies the number of seconds (0-999) before deadman timeout. If this statement is not coded, DEADTIME=60 seconds is the default.

Example

```
DEADTIME=90
```

Related Topics

- The “SET DEADTIME” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DEFBI

The DEFBI statement defines the default input block size for a transport connection. If the user specifies a maximum incoming block size parameter in an SOFFER or SCONNECT request, then the specified size is granted; otherwise, the default is 4096.

The DEFBI statement has the format:

Keyword (Select One)
DEFBI= <i>INPUT-BLOCK-SIZE</i>
DEFBI=4096

INPUT-BLOCK-SIZE

Specifies the default input block size in bytes or Kbytes (for example, DEFBI=4096 or DEFBI=4K). This value can be specified from 2048 to 65535. If this statement is not coded, the default is DEFBI=4096 bytes. Network Executive Software recommends specifying DEFBI=32K.

Example

```
DEFBI=1024
```

Related Topics

- The “SET DEFBI” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DEFBO

The DEFBO statement specifies the default output block size for a transport connection. If the user specifies a maximum outgoing block size parameter in an SOFFER or SCONNECT request, then the specified size is granted; otherwise, the default is 4096.

The DEFBO statement has the format:

Keyword (Select One)
DEFBO= <i>OUTPUT-BLOCK-SIZE</i>
DEFBO=4096

OUTPUT-BLOCK-SIZE

Specifies the default output block size in bytes or Kbytes (for example, DEFBO=4096 or DEFBO=4K). This value can be specified from 2048 to 65535. If this statement is not coded, the default is DEFBO=4096 bytes. Network Executive Software recommends specifying DEFBO=32K.

Example

```
DEFBO=1024
```

Related Topics

- The “SET DEFBO” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DESCCD

The DESC CD statement specifies the default message descriptor code used when NetEx/IP issues messages to the console (WTO messages). The default descriptor code is 5, which indicates an Informational message. Refer to any IBM System Messages manual for a description of Message Descriptor Codes.

The DESC CD statement has the format:

Keyword (Select One)
DESC CD=MESSAGE-DESCRIPTOR-CODE
DESC CD=5

MESSAGE-DESCRIPTOR-CODE

Specifies the message Descriptor Code used by NetEx/IP messages. This number must be specified in the range of 1-16.

Example

DESC CD=6

Related Topics

- The “SET DESC CD” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DFRT

The DFRT statement specifies the starting throughput rate at which NetEx/IP will attempt to deliver data to the network for each network connection. This value is only used for connections to hosts that do not have a ‘rate’ value specified in the PAM or NRB.

The DFRT statement has the format:

Keyword (Select One)
DFRT=DEFAULT-START-RATE
DFRT=0

DEFAULT-START-RATE

Specifies the starting throughput rate for a NetEx/IP connection. If zero is specified, data will be delivered with no internal throttling. This value is specified in Kbits per second. For example, a value of 50 means 50Kbs; a value of 50000 means 50Mbs (i.e. 50,000 Kbps).

Example

DFRT=20

Related Topics

- The “SET DFRT” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “MKBS” initialization statement on page 104

DIRE

The DIRE statement specifies the delay increment receive equivalence value, which is used in protocol-4 throughput calculations. The value specified represents a percentage multiplied by a factor of 10.

The DIRE statement has the format:

Keyword (Select One)
DIRE= <i>DELAY - INCREMENT - RECEIVE - EQUIVALENCE - PERCENTAGE</i>
DIRE=750

DELAY-INCREMENT-RECEIVE-EQUIVALENCE-PERCENTAGE

Specifies the delay increment percentage when receiving data. If this statement is not coded, the default is DIRE=750.

Example

DIRE=800

Related Topics

- The “SET DIRE” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DISE

The DISE statement specifies the delay increment send equivalence value, which is used in protocol-4 throughput calculations. The value specified represents a percentage multiplied by a factor of 10.

The DISE statement has the format:

Keyword (Select One)
DISE= <i>DELAY - INCREMENT - SEND - EQUIVALENCE - PERCENTAGE</i>
DISE=750

DELAY-INCREMENT-SEND-EQUIVALENCE-PERCENTAGE

Specifies the delay increment percentage when sending data. If this statement is not coded, the default is DISE=750.

Example

DISE=800

Related Topics

- The “SET DISE” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

DOWD

The DOWD statement specifies the one-way propagation delay of the network, expressed in milliseconds, and is used if there is no delay value specified in the PAM for a network path. For Type 2 protocol connections, this value represents a fixed propagation delay that never changes. For Type 4 protocol connections, this value represents a starting point that is used for internal bandwidth capacity calculations. The delay is continuously measured during each session, and if it changes, the updated value is used for subsequent internal bandwidth capacity calculations.

The DOWD statement has the format:

Keyword (Select One)
DOWD= <i>ONE-WAY DELAY</i>
DOWD= <u>0</u>

ONE-WAY DELAY

Specifies the one-way delay, in milliseconds, of a network path. A value of 1 millisecond represents the approximate propagation delay of a 100-mile path.

Example

```
DOWD=20
```

Related Topics

- The “SET DOWD” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “MKBS” initialization statement on page 104

DQHB

The DQHB statement specifies the high threshold value (in bytes) for the size of the receiving NetEx DataQue for each network connection. If the size of the Data Queue exceeds this value, subsequent blocks received on the network that are greater than the highest received LRN are NAK'ed and discarded. Subsequent blocks will continue to be discarded until the size of the DataQue is reduced to the value specified by 'DQLB'.

The DQHB statement has the format:

Keyword (Select One)
DQHB=DATAQUE-HIGH WATERMARK
DQHB=20000000

DATAQUE-HIGH WATERMARK

Specifies the high watermark value (in bytes) for the size of a read dataque. Exceeding this value will result in data blocks being discarded, and blocks being resent.

Example

DQHB=30000000

Related Topics

- The “SET DQHB” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “DQLB” initialization statement on page 89

DQLB

The DQLB statement specifies the low threshold value (in bytes) for the size of the receiving NetEx DataQue for each network connection. See the 'DQHB' parameter for the description of how this value is used.

The DQLB statement has the format:

Keyword (Select One)
DQLB= <i>DATAQUE-LOW WATERMARK</i> DQLB=10000000

DATAQUE-LOW WATERMARK

Specifies the low watermark value (in bytes) for the size of a read dataque. This value is only used if the size of a read dataque exceeds the value specified by DQHB.

Example

DQLB=20000000

Related Topics

- The “SET DQLB” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “DQHB” initialization statement on page 88

DRIVAVG

The DRIVAVG statement specifies the estimated average number of NetEx Driver service sessions that is active for this host. NetEx uses this value to allocate memory for control blocks.

The DRIVAVG statement has the format:

Keyword (Select One)
DRIVAVG= <i>NUMBER-OF-SESSIONS</i> DRIVAVG=2

NUMBER-OF-SESSIONS

Specifies the average number of sessions (2 or greater). If this statement is not coded, the default is DRIVAVG=2.

Example

DRIVAVG=4

Related Topics

- “DRIVLIM” initialization statement on page 90
- “DRIVMAX” initialization statement on page 91

DRIVLIM

The DRIVLIM statement specifies the upper limit of the DRIVMAX parameter. Note that the DRIVMAX value is operator-changeable, but DRIVLIM is set at initialization and is an absolute limit.

The DRIVLIM statement has the format:

Keyword (Select One)
DRIVLIM= <i>NUMBER-OF-SESSIONS</i>
DRIVLIM=8

NUMBER-OF-SESSIONS

Specifies the maximum number of sessions (2 or greater). If this statement is not coded, the default is DRIVLIM=8.

Example

DRIVLIM=20

Related Topics

- “DRIVAVG” initialization statement on page 89
- “DRIVMAX” initialization statement on page 91

DRIVMAX

The DRIVMAX statement specifies the maximum number of concurrent Driver service connections supported by NetEx on this system. The count of connections is equivalent to the number of subchannels allocated to Driver service users. If a new DCONNECT request is made when DRIVMAX connections have been made, the request is rejected with an error.

The DRIVMAX statement has the format:

Keyword (Select One)
DRIVMAX=NUMBER-OF-SESSIONS
DRIVMAX=8

NUMBER-OF-SESSIONS

Specifies the maximum number of Driver service sessions (from 0 to the DRIVLIM value). If this statement is not coded, the default is DRIVMAX=8.

Example

```
DRIVMAX=10
```

Related Topics

- The “SETDRIVMAX” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “DRIVAVG” initialization statement on page 89
- “DRIVLIM” initialization statement on page 90

ENDINI1

The ENDINI1 statement terminates initialization statement processing, and allows subsequent NetEx/IP operator commands to be inserted in the NTXINIT initialization file.

The ENDINI1 statement has the following format:

Keyword (No Operands)
ENDINI1

Example

```
HOST NAME=HOST1
(Remaining initialization file statements)
ENDINI1
* Start of Operator Commands to be executed after initialization
SET IP 0200 10.1.5.100
SET PRINT ON
D L
```

FMSML

The FMSML statement specifies whether or not to force multiline WTOs on remote operator displays. If OFF, the display will use single line WTOs for the display if any line is greater than 70 characters.

The FMSML statement has the format:

Keyword (Select One)
FMSML=ON
FMSML=OFF

ON

Will force multiline WTOs for remote operator displays, which will truncate the display if any lines are greater than 70 characters.

OFF

Will use single line WTOs for remote operator displays if any line is greater than 70 characters. Those lines greater than 70 characters will wrap in the display. The default is OFF.

Related Topics

- The “SET FMSML” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

FUBS

The FUBS statement specifies whether or not to free unused buffer space within allocated buffers. Normally, this value should be specified as OFF. Enabling this feature will result in buffer memory fragmentation.

The FUBS statement has the format:

Keyword (Select One)
FUBS=ON
FUBS=OFF

ON

Enables freeing unused memory within allocated buffers.

OFF

Disables freeing unused memory within allocated buffers. The default is OFF.

Related Topics

- The “SET FUBS” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

GTRACE

The GTRACE statement enables/disables the NetEx/IP GTF trace mechanism.

The GTRACE statement has the format:

Keyword (Select One)

GTRACE=ON GTRACE=OFF

ON

Enables NetEx/IP user record (type USR) GTF tracing. At specific processing points, NetEx/IP issues the GTRACE macro to “MC” (Monitor Call) put USR data to GTF. The USR entries will appear as hexadecimal-format entries in a GTF data set. The specific USR Event Identifiers (EID) used by NetEx/IP are listed in the description of the NetEx/IP SET GTRACE command in the *H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual*.

NOTE: GTF must be active, tracing the NetEx/IP address space, and enabled for “USR” event recording, or optionally by USRP recording for specific EIDs. GTRACE recording should only be used when requested by NetEx support personnel..

OFF

Disables NetEx/IP user record GTF tracing. The default is OFF.

Related Topics

- The “SET GTRACE” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

HOST NAME

The HOST NAME statement identifies this active NetEx’s host name. The name specified must exactly match the local HOST NAME specified on the Configuration Manager HOST statement This is the name used in CONNECT requests.

The HOST NAME statement has the format:

Keyword (Select One)

HOST NAME=HOSTNAME

HOST NAME=ZOS

HOSTNAME

Specifies a one to eight character host name. If this statement is not coded, the default is HOST NAME=ZOS.

Example

HOST NAME=CLEMENS

Related Topics

- None.

IDLEMAX

The IDLEMAX statement specifies the maximum interval between idle messages if using the inactive idle scheme (INACTIVE=ON).

The IDLEMAX statement has the format:

Keyword (Select One)
IDLEMAX=IDLE-TIMEOUT
IDLEMAX=240

IDLE-TIMEOUT

Specifies the maximum idletime in seconds (1-999). If this statement is not coded, the default is IDLEMAX=240.

Example

IDLEMAX=300

Related Topics

- The “SET IDLEMAX” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “IDLETIME” initialization statement on page 94

IDLETIME

The IDLETIME statement specifies the number of seconds that Transport protocol waits before sending an “idle” message to verify the continued existence of the communicating party at the other end of the connection. The time between idle messages are specified in seconds.

The IDLETIME statement has the format:

Keyword (Select One)
IDLETIME=IDLE-TIMEOUT
IDLETIME=5

IDLE-TIMEOUT

Specifies the number of seconds (0-999) before a idle timeout occurs. If this statement is not coded, the default is IDLETIME=5 seconds.

Example

IDLETIME=10

Related Topics

See the following sections for more information:

- The “SET IDLETIME” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “IDLEMAX” initialization statement on page 93

INACTIVE

The INACTIVE statement specifies whether to use the inactive idle scheme or not. If set ON, the idle messages are sent less frequently the longer the session remains inactive (data not being sent or received). This is a useful feature for sessions that stay connected indefinitely but only send data occasionally. This option reduces the idle traffic between sessions from a minimum of IDLETIME to a maximum of IDLEMAX.

The INACTIVE statement has the format:

Keyword (Select One)
INACTIVE=ON
INACTIVE=OFF

ON

Selects the inactive idle scheme for all sessions.

OFF

Selects the normal idle scheme for all sessions. If this statement is not coded, the default is INACTIVE=OFF.

Example

```
INACTIVE=ON
```

Related Topics

- The “SET INACTIVE” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

IOCOUNT

The IOCOUNT statement specifies whether to enable or disable tracking of I/O activity performance events. This parameter is used for SMF record reporting.

The IOCOUNT statement has the format:

Keyword (Select One)
IOCOUNT=ON
IOCOUNT=OFF

ON

Initializes tracking relative to I/O activity performance events.

OFF

Disables tracking relative to I/O activity performance events. If this statement is not coded, the default is IOCOUNT=OFF.

Related Topics

See the following sections for more information on SMF record reporting:

- “SMF Record Layout” on page 17
- “SMFTYPE” initialization statement on page 125
- “XDISC” initialization statement on page 145

LARNAME

The LARNAME statement specifies the linkage-assist module name that resides in LPALIB. A maximum of 8 characters are allowed.

The LARNAME statement has the format:

Keyword (Select One)
LARNAME= <i>LAR-NAME</i>
LARNAME=NXMLAR

LAR-NAME

Specifies the one to eight character linkage-assist module name. If this statement is not coded, the default is LARNAME=NXMLAR.

Example

LARNAME=SAWYER

LOGCMD

The LOGCMD statement specifies whether to log all NetEx/IP operator commands into the NTXLOG SYSOUT file. If set ON, all NetEx/IP operator commands are logged into NTXLOG.

The LOGCMD statement has the format:

Keyword (Select One)
LOGCMD=ON
LOGCMD=OFF

ON

Enables the logging of all NetEx operator commands into NTXLOG.

OFF

Disables the logging of all NetEx operator commands into NTXLOG.

Example

```
LOGCMD=ON
```

Related Topics

- The “SET LOGCMD” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXBI

The MAXBI statement specifies the maximum number of data bytes that may be received in any session read. If a session read specifies a buffer length greater than MAXBI, the request will be rejected. The MAXBI statements reduces fragmentation or excessive use of NetEx buffer memory.

The MAXBI statement has the format:

Keyword (Select One)
MAXBI=MAXIMUM-BLOCK-SIZE
MAXBI=61440

MAXIMUM-BLOCK-SIZE

Specifies the maximum input block size in bytes or Kbytes (for example, MAXBI=4096 or MAXBI=4K). This value can be specified from 2048 to 65535. If this statement is not coded, the default is MAXBI=61440 bytes.

Example

```
MAXBI=1024
```

Related Topics

- The “SET MAXBI” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXBO

The MAXBO statement specifies the maximum number of data bytes that may be sent on any session write request. If session write specifies a buffer length greater than MAXBO, the request will be rejected. The MAXBO statement reduces fragmentation or excessive use of NetEx buffer memory.

The MAXBO statement has the format:

Keyword (Select One)
MAXBO=MAXIMUM-BLOCK-SIZE
MAXBO=61440

MAXIMUM-BLOCK-SIZE

Specifies the maximum output block size in bytes or Kbytes (for example, MAXBI=4096 or MAXBI=4K). This value can be specified from 2048 to 65535. If this statement is not coded, the default is MAXBO=61440 bytes.

Example

MAXBO=1K

Related Topics

- The “SET MAXBO” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXDDBQ

The MAXDDBQ statement specifies the maximum number of Driver Interface Data Blocks (DDBs) that can be queued before incoming DDBs are rejected.

The MAXDDBQ statement has the format:

Keyword (Select One)
MAXDDBQ=NUMBER-OF-DATABLOCKS
MAXDDBQ=6

NUMBER-OF-DATABLOCKS

Specifies the maximum number of Driver Interface Data Blocks (DDBs) that can be queued. If this statement is not coded, the default is MAXDDBQ=6.

Example

MAXDDBQ=10

Related Topics

- The “SET MAXDDBQ” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXDREAD

The MAXDREAD statement specifies the maximum number of concurrent DREAD requests for a given connection.

The MAXDREAD statement has the format:

Keyword (Select One)
MAXDREAD=NUMBER-OF-DREADS MAXDREAD=2

NUMBER-OF-DREADS

Specifies the maximum number of DREADs for a given connection. If this statement is not coded, the default is MAXDREAD=2.

Example

MAXDREAD=

Related Topics

- The “SET MAXDREAD” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXDWRIT

The MAXDWRIT statement specifies the maximum number of concurrent DWRITE requests for a given connection.

The MAXDWRIT statement has the format:

Keyword (Select One)
MAXDWRIT=NUMBER-OF-DWRITES MAXDWRIT=2

NUMBER-OF-DWRITES

Specifies the maximum number of DWRITES (0-32767) for a given connection. If this statement is not coded, the default is MAXDWRIT=2.

Example

MAXDWRIT=4

Related Topics

- The “SET MAXDWRIT” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXODATA

The MAXODATA statement specifies the maximum size of the ODATA segments that are included in the message proper actually sent on the network by this host. The NetEx software divides the blocks of ODATA it receives from the user into segments, transmits them across the network as part of the message proper at that size, and reconstructs the data on the receiving side to the appropriate block size. Note that this value may be less than the MAXBI/MAXBO value without affecting the user program.

The MAXODATA statement has the format:

Keyword (Select One)
MAXODATA=MAXIMUM-ODATA-SEGMENT-SIZE
MAXODATA=3072

MAXIMUM-ODATA-SEGMENT-SIZE

Specifies the maximum ODATA segment size in bytes or Kbytes (for example, MAXBI=1024 or MAXBI=1K). This value can be specified from 64 to 32603. If this statement is not coded, the default is MAXODATA=3072 bytes.

Example

MAXODATA=1024

Related Topics

- The “SET MAXODATA” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MAXSEG

The MAXSEG statement specifies the maximum size of the data segments that are actually sent on the network by this host. The NetEx software divides the blocks of data it receives from the user into segments, transmits them across the network at that size, and reconstructs the data on the receiving side to the appropriate block size. Note that this value may be less than the MAXBI/MAXBO value without affecting the user program.

The maximum segment size is closely related to the segment size defined in the NTCROUTE installation macro. If no segment size is specified for a route, the default is the MAXSEG size defined for the host issuing the NTCROUTE statement. The MAXSEG is also an upper limit for the segment size specified in the NTCROUTE statement (that is, the lower of the two values is used for that route).

The MAXSEG statement has the format:

Keyword (Select One)
MAXSEG=MAXIMUM-SEGMENT-SIZE
MAXSEG=32767

MAXIMUM-SEGMENT-SIZE

Specifies the maximum segment size in bytes or Kbytes from 1K to 65535 bytes. The minimum allowed segment size is 1024 and the maximum allowed segment size is 65535. For example, MAXSEG=1024 or MAXSEG=1K. If this statement is not coded, the default is MAXSEG=32767 bytes. Network Executive Software recommends specifying MAXSEG=65535.

NOTE: If MAXSEG needs to be set to a low value (i.e. to avoid fragmentation) set MAXODATA to a value 164 bytes below the desired MAXSEG.

Example

MAXSEG=1024

Related Topics

- The “SET MAXSEG” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MBFI

The MBFI statement specifies the maximum number of input blocks that NetEx/IP will allow to be outstanding for each user. This may have to be increased for long latency or high bandwidth networks, particularly if small segment sizes are specified (e.g. if maxseg is specified as 4K or less). If your network falls into this category, please contact support@netex.com for guidance on setting proper values for this parameter.

CAUTION: Usage of this initialization statement requires H210IPZ to be at the Release 7.0.5 level or higher.

The MBFI statement has the format:

Keyword (Select One)
MBFI=MAXIMUM-NUMBER-OF-INBOUND-DATABLOCKS
MBFI=5

MAXIMUM-NUMBER-OF-INBOUND-DATABLOCKS

Specifies the maximum number of inbound datablocks. This parameter is only used for protocol-2 connections. A value larger than 5 should be specified for high bandwidth and long latency networks. If this statement is not coded, the default is MBFI=5

Example

MBFI=10

Related Topics

- The “SET MBFI” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MBFO

The MBFO statement specifies the maximum number of output blocks that NetEx/IP will allow to be outstanding for each user. This may have to be increased for long latency or high bandwidth networks, particularly if small segment sizes are specified (e.g. if maxseg is specified as 4K or less). If your network falls into this category, please contact support@netex.com for guidance on setting proper values for this parameter.

CAUTION: Usage of this initialization statement requires H210IPZ to be at the Release 7.0.5 level or higher.

The MBFO statement has the format:

Keyword (Select One)
MBFO= MAXIMUM-NUMBER-OF-OUTBOUND-DATABLOCKS
MBFO=5

MAXIMUM-NUMBER-OF-OUTBOUND-DATABLOCKS

Specifies the maximum number of outbound datablocks. This parameter is only used for protocol-2 connections. A value larger than 5 should be specified for high bandwidth and long latency networks.

If this statement is not coded, the default is MBFO=5.

Example

MBFO=12

Related Topics

- The “SET MBFO” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MFCP

The MFCP statement specifies the Memory Free Copy ratio, and is used to determine how to process buffer size requirements for data coming in from the network. When this ratio is set, it specifies the denominator of the fraction –

$1/\text{mfcp}$

When MFCP is specified as a value between 2 and 16, and the amount of incoming data is less than or equal to $(1/\text{mfcp} \times \text{SEGSIZE})$, then a new buffer of the exact size required to hold the data is obtained, the data is copied into it, and the original buffer is freed in its entirety. If the amount of incoming data is greater than $(1/\text{mfcp} \times \text{SEGSIZE})$, then the data is not copied into a smaller buffer, but the unused space in the original buffer is freed.

If MFCP is specified as zero, then the incoming data is never copied into smaller buffers, regardless of size, but any unused buffer space within the original buffer is freed.

MFCP can be used to prevent fragmentation of NetEx buffers. It has the most benefit when there are significant numbers of small buffers on the network intermixed with large buffers.

The MFCP statement has the format:

Keyword (Select One)
MFCP=MEMORY-FREE-COPY
MFCP=0

MEMORY-FREE-COPY

Specifies the denominator of the fraction “ $1/\text{mfcp}$ ”. When an incoming data buffer is less than or equal to $(1/\text{mfcp} \times \text{SEGSIZE})$, data is copied into a new buffer of the correct size. When used, this value should be specified between 2 and 16. If 0 is specified, data is not copied.

Example

MFCP=4

This causes incoming data to be copied into a smaller buffer whenever the amount of actual data is less than or equal to $\frac{1}{4}$ of the size of the original buffer (i.e. $\frac{1}{4}$ of SEGSIZE).

Related Topics

- The “SET MFCP” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MIHTM

The MIHTM command specifies the time interval (in seconds) that the NetEx missing interrupt handler waits between checks of the device blocks. On the average, if an I/O does not complete within MIHTM + MIHTM/2, the I/O is halted.

The MIHTM statement has the format:

Keyword (Select One)
MIHTM= <i>SECONDS</i>
MIHTM=15

SECONDS

Specifies the NetEx missing interrupt handler interval in seconds. Allowed values are 0-32767; the default is 15. Any new interval value takes effect only after the previous interval completes. If MIHTM is set to zero, the missing interrupt handler exits, and can only be restarted by bringing up NetEx again.

Example

MIHTM=20

Related Topics

- The “SET MIHTM” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MKBS

The MKBS initialization statement specifies the maximum rate at which NetEx/IP will deliver data to the network for each network connection. If zero is specified, data will be delivered with no internal throttling. This value is only used for connections to hosts that do not have a ‘rate’ value specified in the PAM. This value is specified in Kbits per second. For example, a value of 50 means 50Kbs; a value of 50000 means 50Mbps (i.e. 50,000 Kbs).

The MKBS statement has the format:

Keyword (Select One)
MKBS= <i>RATE</i>
MKBS=0

RATE

Specifies the maximum rate, in kilobits per second, at which NetEx/IP will deliver data to the network. For Type 4 protocol connections, the rate will be dynamically adjusted to match the network capability.

Example

MKBS=100000

Related Topics

- The “SET MKBS” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

MSGLVL

The MSGLVL statement specifies message level (0-15 decimal). All NetEx messages generated internally are given a priority from 0 to 15. If the message priority is equal to or greater than the current MSGLVL, the message is printed; otherwise, it is ignored. There are five basic categories of priority:

- 12 - Messages that require immediate action by the operator.
- 8 - Messages that are of great interest to the operator and may require operator action. The default is 8.
- 6 - Messages that are normally considered to be a non-permanent error type.
- 4 - Messages of events that are of interest in closely monitored environments.
- 0 - Messages that are intended for diagnostic or debugging purposes. These messages are generally only of interest when a system programmer is attempting to diagnose a NetEx problem.

Refer to the SET MSGLVL operator command for related information. The route code is also used to control receiving messages (refer to the ROUTCD initialization statement for more information).

The MSGLVL statement has the format:

Keyword (Select One)
MSGLVL= <i>LEVEL</i> MSGLVL=8

LEVEL

Specifies the message level (0-15 decimal). If this statement is not coded, the default is MSGLVL=8.

Example

MSGLVL=12

Related Topics

- The “SET MSGLVL” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “MSGML” initialization statement on page 106
- “MSGTM” initialization statement on page 107
- “MSGXL” initialization statement on page 108
- See the H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages and Abend Codes manual for more information on NetEx messages.

MSGML

The MSGML statement specifies if NetEx issues Write To Operator (WTO) messages in multiple or single line format.

The MSGML statement has the format:

Keyword (Select One)
MSGML=ON
MSGML=OFF

ON

Enables multiple line format for these messages. If this statement is not coded, the default is MSGML=ON.

OFF

Enables in single line format for these messages.

Example

```
MSGML=OFF
```

Related Topics

- The “SET MSGML” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “MSGLVL” initialization statement on page 105
- “MSGTM” initialization statement on page 107
- “MSGXL” initialization statement on page 108
- See the H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages and Abend Codes manual for more information on NetEx messages.

MSGTM

The MSGTM statement specifies whether timestamp precedes messages. If MSGTM=ON, the first line of each message is preceded by the time. The format of the timestamp is HH.MM.SS.

The MSGTM statement has the following format:

Keyword (Select One)
MSGTM=ON
MSGTM=OFF

ON

Inserts a timestamp before the NetEx messages.

OFF

Displays messages without timestamps. If this statement is not coded, the default is MSGTM=OFF.

Example

```
MSGTM=ON
```

Related Topics

- “SET MSGTM” command in H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual
- “MSGLVL” initialization statement on page 105
- “MSGML” initialization statement on page 106
- “MSGXL” initialization statement on page 108
- See the H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages and Abend Codes manual for more information on NetEx messages.

MSGXL

The MSGXL statement specifies if messages is translated into uppercase. In MSGXL=OFF, messages appears in mixed case.

The MSGXL statement has the format:

Keyword (Select One)
MSGXL=ON
MSGXL=OFF

ON

Displays messages in uppercase.

OFF

Displays messages in mixed-case. If this statement is not coded, the default is MSGXL=OFF.

Example

```
MSGXL=ON
```

Related Topics

- The “SET MSGXL” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “MSGLVL” initialization statement on page 105
- “MSGML” initialization statement on page 106
- “MSGTM” initialization statement on page 107
- See the H210IPZ NetEx/IP for IBM z/OS Operating Systems Messages and Abend Codes manual for more information on NetEx messages.

MULTIH

The MULTIH statement (multihost) is used with the TNP feature. When set to ON, the NetEx/IP Requester hostname that is specified in the Requester configuration file is used as the NetEx/IP session hostname. Typically, MULTIH should be set to ON in TNP configurations. Refer to the section “

Adding NetEx/IP Requester Hosts to the NCT” on page 36 for a description of MULTIH and TNP.

The MULTIH statement has the format:

Keyword (Select One)
MULTIH=ON
MULTIH=OFF

ON

Sets the MULTIH feature on. This allows NetEx to distinguish between SOFFERs of the same application name when using NetEx/IP Requester hosts with TNP.

OFF

Sets the MULTIH feature off. The host names of the NetEx/IP Requester hosts are not used.

Example

```
MULTIH=ON
```

Related Topics

- The “SET MULTIH” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “MSGLVL” initialization statement on page 105
- “MSGML” initialization statement on page 106
- “MSGTM” initialization statement on page 107

NTXOPER

The NTXOPER statement disables/enables the remote operator service. This is a useful facility for interrogating a remote NetEx to return displays or to set values remotely. This statement may also specify the class of operator commands that the remote operator is allowed to issue.

There is one remote operator session for every NTXOPER statement included.

The NTXOPER statement has the format:

Keyword (Select One)	
NTXOPER=ON	CLASS=A
NTXOPER=ON	CLASS=C
NTXOPER=ON	CLASS=G
NTXOPER=OFF	

ON

Enables remote operator service.

CLASS (use only if ON is selected)

This is the level of operator command that is accepted from the remote host. Class may also be selected using the SET ROPCLASS command.

G =

General class commands (DISPLAY).

C =

Privileged class commands (SET, START, > (remote command)) and general class. Network Executive Software recommends specifying CLASS=G.

A =

All commands (HALT, KILL, DRAIN included).

OFF

Disables remote operator service for all remote operators. If this statement is not coded, the default is NTXOPER=OFF.

Example

NTXOPER=ON CLASS=C

Related Topics

- The “SET NTXOPER” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

OSAA

The OSAA statement identifies a local ADAPTER defined in the Network Control Table (NCT) as an Open Systems Adapter (OSA) interface. The value specified corresponds to the NETADDR and SMGDREF of a local ADAPTER defined in the NCT. All communications to and from an OSA interface is done by using the onboard TCP/IP stack.

Note: Support of the OSA feature is separately licensed, and enabled by a software key.

The OSAA statement has the format:

Keyword
OSAA=UUSS [IPADDR]

UUSS

UU is the NETADDR value and SS is the SMGDREF value of a local ADAPTER defined in the NCT to use as an OSA interface.

IPADDR (optional)

This is the IP address which will be mapped to the UUSS for this local ADAPTER. If not specified, the IP address will be obtained via DNS. If specified on any OSAA statement, the OSA1IF statement should be set to OFF.

Related Topics

- See “OSA1IF” for more information.

OSA1IF

The OSA1IF statement indicates whether NetEx/IP should bind to a specific IP address (OFF), or to an IP address of 0 (ON). If ‘ON’ is set, this NetEx/IP can receive any NetEx/IP messages, even those that don’t map to the local OSAA definition. If more than 1 OSAA statement is defined, then OSA1IF should always be set to OFF. If this parameter is not specified, the default is ON.

The OSA1IF statement has the format:

Keyword (Select One)
OSA1IF=OFF
OSA1IF=ON

ON

NetEx/IP binds to an IP address of 0, allowing NetEx/IP traffic to be received on any NetEx/IP OSA interface.

OFF

NetEx/IP binds to the specific IP address of the OSA device, restricting NetEx/IP traffic to only be received on the NetEx/IP OSA interface corresponding to the inbound IP address. The default is ON.

Related Topics

- See “OSAA” initialization statement for more information.

OUTCLASS

The OUTCLASS statement specifies the SYSOUT class.

The OUTCLASS statement has the format:

Keyword (Select One)
OUTCLASS=A
OUTCLASS=X

X

Specifies any valid JES output class. The default is A.

Related Topics

- The “SET OUTCLASS” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- The “SET OUTLIM” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

PAGENIT

The PAGENIT statement allocates a specified number of extra 4K pages of memory to hold NITs, the central control blocks for NetEx Internal Tasks. Normally, NetEx initialization calculates a default number of pages for NITs based on the SESLIM statement specified during initialization. (For example, if SESLIM=32, 16 pages are allocated, in addition to the number specified by the PAGENIT statement.) If special considerations require extra NITs during NetEx execution, one or more 4K pages of NITs may be allocated through this PAGENIT statement. The number of pages specified using this parameter is added to the number calculated by NetEx during initialization. Approximately 64 NITs fit in a single 4K page.

The PAGENIT statement has the format:

Keyword (Select One)
PAGENIT=PAGES
PAGENIT=0

PAGES

Specifies the number of extra 4K pages (0-2048). If this statement is not coded, the default is PAGENIT=0 pages.

Example

```
PAGENIT=8
```

Related Topics

- “PAGESAVE” initialization statement on page 113
- “PAGEXTRA” initialization statement on page 114
- “PAGExxx” initialization statement on page 115

PAGESAVE

The PAGESAVE statement allocates the specified number of extra 4K pages of register save areas that are used during NetEx internal subroutine calls. Each internal call allocates a save area that is preserved until return is made to the calling module. If NetEx has insufficient save areas, it abends. Normally, NetEx initialization calculates a default number of pages for save areas based on the SESLIM statement specified during initialization. (For example, if SESLIM=32, 65 pages are allocated, in addition to the number specified by the PAGESAVE statement.) If special considerations require more save areas during execution, one or more extra 4K pages of save areas may be allocated during initialization with the PAGESAVE statement. The number of pages specified using this statement is added to the number calculated by NetEx during initialization.

The PAGESAVE statement has the format:

Keyword (Select One)
PAGESAVE= <i>PAGES</i>
PAGESAVE= <u>100</u>

PAGES

Specifies the number of extra 4K pages (0-1800). If this statement is not coded, the default is PAGESAVE=100.

Example

PAGESAVE=4

Related Topics

- “PAGENIT” initialization statement on page 112
- “PAGEXTRA” initialization statement on page 114
- “PAGExxx” initialization statement on page 115

PAGEXTRA

The PAGEXTRA statement specifies the number of additional 4K pages of memory that are available for the overflow of the 16, 32, 64, 128, 256, or 512-byte page areas. The total amount of extra memory from any combination of these page areas may not exceed the PAGEXTRA value. Once this limit is reached, additional requests for storage is queued.

For more information about NetEx pages, refer to the PAGENIT and PAGExxx statements in this section.

The PAGEXTRA statement has the format:

Keyword (Select One)
PAGEXTRA= <i>PAGES</i>
PAGEXTRA=4

PAGES

Specifies the number of extra 4K pages (0-32767). If this statement is not coded, the default is PAGEXTRA=4 pages.

Example

PAGEXTRA=8

Related Topics

- “PAGENIT” initialization statement on page 112
- “PAGEXTRA” initialization statement on page 114
- “PAGExxx” initialization statement on page 115

PAGExxx

There are ten PAGEXXX statements, each of which allocate a specified number of extra 4K pages of additional memory and data buffers used for miscellaneous control block storage areas by NetEx. The xxx represents the size (in bytes) of the storage area. The PAGEXXX statements are: PAGE16, PAGE32, PAGE64, PAGE128, PAGE256, PAGE512, PAGE1K, PAGE4K, PAGE8K, and PAGE16K.

NetEx initialization provides enough pages for each subpool based on the SESLIM statement and the sizes of certain major NetEx control blocks. The PAGE256, PAGE512, and PAGE1K pools are allocated enough memory so that SESLIM sessions have enough buffer space for local connections. Long latency connections or the use of very large buffers (>8K) may require the user to specify extra pools for these subpools. If extra pages are required, they may be allocated using the PAGExxx statement. The number of pages specified using this parameter is added to the number calculated by NetEx during initialization. Note that each pool of storage requires an integral number of pages. For example, PAGE32=1 would allocate one additional 4K page of 32-byte storage areas (128 32-byte areas) that could not be used by any other subpool.

The PAGExxx statement has the format:

Keyword (Select One)
PAGE16=PAGE\$
PAGE16=0
PAGE32=PAGE\$
PAGE32=0
PAGE64=PAGE\$
PAGE64=0
PAGE128=PAGE\$
PAGE128=50
PAGE256=PAGE\$
PAGE256=0
PAGE512=PAGE\$
PAGE512=0
PAGE1K=PAGE\$
PAGE1K=2000
PAGE4K=PAGE\$
PAGE4K=0
PAGE8K=PAGE\$
PAGE8K=0
PAGE16K=PAGE\$
PAGE16K=0

PAGES

Specifies the number of extra 4K pages.

The valid parameter ranges and default for each PAGEXXX statement are as follows:

<u>Statement</u>	<u>Range</u>	<u>Default</u>	<u>Statement</u>	<u>Range</u>	<u>Default</u>
PAGE16	0-512	0	PAGE512	0-16384	0
PAGE32	0-1024	0	PAGE1K	0-32767	2000
PAGE64	0-2048	0	PAGE4K	0-32767	0
PAGE128	0-4096	50	PAGE8K	0-32767	0
PAGE256	0-8192	0	PAGE16K	0-32767	0

Example

PAGE16=4

Related Topics

- “PAGENIT” initialization statement on page 112
- “PAGEXTRA” initialization statement on page 114
- “PAGExxx” initialization statement on page 115

PDQNAME

The PDQNAME statement specifies the SRB PURGEDQ module name that resides in LPALIB. A maximum of 8 characters are allowed.

The PDQNAME statement has the format:

Keyword (Select One)
PDQNAME= <i>PDQ-NAME</i>
PDQNAME= <u>NXMSRBR1</u>

PDQ-NAME

This required parameters is the one to eight character SRB PURGEDQ module name. If this statement is not coded, the default is PDQNAME=NXMSRBR1.

Example

PDQNAME=YOSEMITE

PRFP

The PRFP statement specifies the default preferred protocol type (2 or 4) to use when connecting to hosts that support multiple NetEx/IP protocol types. Network throughput for protocol type-2 connections is controlled by static parameters, while network throughput for protocol type-4 connections is controlled by dynamic parameters, based on the characteristics of the network.

Caution: both sides of a NetEx/IP connection must be at comparable protocol type-4 levels for the protocol type-4 capability to be supported.

The PRFP statement has the format:

Keyword (Select One)
PRFP= <i>PROTOCOL-TYPE</i>
PRFP= <u>2</u>

PROTOCOL-TYPE

This parameter specifies the protocol type to use for NetEx/IP connections. This value must be specified as either 2 (for protocol-2 connections) or 4 (for protocol-4) connections. If this statement is not coded, the default is PRFP=2.

Example

```
PRFP=2
```

Related Topics

- The “SET PRFP” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

RDDP

The RDDP statement specifies the rate delay decrement percentage, used for protocol type-4 NetEx/IP connections. This is the percentage factor used to decrease the sending rate of a network connection if the round-trip delay increases. This recalculation is performed after the expiration of each interval specified by the RDII parameter. The value specified represents a percentage multiplied by a factor of 10.

The RDDP statement has the format:

Keyword (Select One)
<i>RDDP=RATE-DELAY-DECREMENT-PERCENTAGE</i> <i>RDDP=250</i>

RATE-DELAY-DECREMENT-PERCENTAGE

Specifies the rate delay decrement percentage, expressed as a percentage multiplied by a factor of 10. For example, a value of 250 indicates a value of 25.0%. If this statement is not coded, the default is RDDP=250.

Example

RDDP=100

Related Topics

- The “SET RDDP” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

RDII

The RDII statement specifies the round trip delay increase interval (in seconds), used for protocol type-4 NetEx/IP connections. This is the interval (in seconds) used by the sending side of a network connection after which a check is made for an increase in the round trip delay.

The RDII statement has the format:

Keyword (Select One)
<i>RDII=ROUNDTrip-DELAY-INCREASE-INTERVAL</i> <i>RDII=60</i>

ROUNDTrip-DELAY-INCREASE-INTERVAL

Specifies the number of seconds (0-999) that must transpire before a roundtrip time will be increased because of a new calculation. If this statement is not coded, the default is RDII=60.

Example

RDII=10

Related Topics

- The “SET RDII” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

READTIME

The READTIME statement specifies the number of seconds that transport software retains data if the user does not issue a READ request to accept incoming data, Connect Indications, or Disconnect Indications. If Transport detects that data will be sent to the user, it waits for the specified READTIME seconds. If the data is not sent by that time, then the transport software issues a Disconnect to the corresponding party. All waiting data in the local host that was intended for the timed-out application is discarded.

If the application that failed to issue a READ does issue a READ after this time, it receives the “T-Ref invalid” response indicating that Transport has no record of the transport connection. For example, this condition occurs during the time required to locate and mount tapes.

NOTE: When coding this statement, be sure to consider tape mounting conditions so that data does not time out while a tape is waiting to be mounted.

The READTIME statement has the format:

Keyword (Select One)
READTIME= <i>READ-TIMEOUT</i>
READTIME=60

READ-TIMEOUT

Specifies the number seconds (1-9999) before timeout. Specifying zero could produce an unpredictable result. If this statement is not coded, the default is READTIME=60 seconds. Network Executive Software recommends specifying READTIME=300.

Example

```
READTIME=90
```

Related Topics

- The “SET READTIME” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

ROUTCD

The ROUTCD statement specifies the WTO route code used for NetEx operator messages. This statement allows users to control which operator consoles receive NetEx messages in the same way that other types of messages are controlled.

The ROUTCD statement has the format:

Keyword (Select One)
ROUTCD= <i>ROUTE-CODES</i>
ROUTCD=2

ROUTE-CODES

Specifies the route codes for NetEx messages (0-16). If multiple route codes are specified, they must be separated by a comma. If this statement is not coded, the default is ROUTCD=2.

Example

ROUTCD=4

Related Topics

- The “SET ROUTCD” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

RQPH

The RQPH statement specifies the receive rate percentage high, used for protocol type-4 NetEx/IP connections. This value represents the high bound of the send and receive equivalence adjustment (see RQPS). The value specified represents a percentage multiplied by a factor of 10.

The RQPH statement has the format:

Keyword (Select One)
RQPH= <i>RECEIVE-RATE-PERCENTAGE-HIGH</i>
RQPH=855

RECEIVE-RATE-PERCENTAGE-HIGH

Specifies the high bound of the send and receive equivalence percentage. If this statement is not coded, the default is RQPH=855.

Example

RQPH=600

Related Topics

- The “SET RQPH” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

RQPL

The RQPL statement specifies the receive rate percentage low, used for protocol type-4 NetEx/IP connections. This value represents the low bound of the send and receive equivalence adjustment (see RQPS). The value specified represents a percentage multiplied by a factor of 10.

The RQPL statement has the format:

Keyword (Select One)
RQPL= <i>RECEIVE-RATE-PERCENTAGE-LOW</i> RQPL=500

RECEIVE-RATE-PERCENTAGE-LOW

Specifies the low bound of the send and receive equivalence percentage. If this statement is not coded, the default is RQPL=500.

Example

RQPL=200

Related Topics

- The “SET RQPL” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

RQPS

The RQPS statement specifies the receive rate percentage equivalence, used for protocol type-4 NetEx/IP connections. This value represents the percentage factor used to determine equivalence of the send and receive rates for each network connection. These rates are assumed to be equal if they fall within this percentage of each other. This is the initial value used for each network connection, and can be dynamically adjusted based on activity and performance of the network. The value specified represents a percentage multiplied by a factor of 10.

The RQPS statement has the format:

Keyword
RQPS= <i>RECEIVE-RATE-PERCENTAGE-EQUIVALENCE</i> RQPS=750

RECEIVE-RATE-PERCENTAGE-EQUIVALENCE

Specifies the send and receive equivalence percentage. If this statement is not coded, the default is RQPS=750.

Example

RQPS=500

Related Topics

- The “SET RQPS” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

RRCI

The RRCI statement specifies the receive rate calculation interval (in seconds) , used for protocol type-4 NetEx/IP connections. This value represents the time interval (in seconds) after which the receive rate for a network connection is recalculated. The receive rate refers to the rate at which the application is receiving data from NetEx.

The RRCI statement has the format:

Keyword (Select One)
<i>RRCI=RECEIVE-RATE-CALCULATION-INTERVAL</i> <i>RRCI=2</i>

RECEIVE-RATE-CALCULATION-INTERVAL

Specifies the number of seconds (0-999) that must transpire, after which time the receive rate for a network connection is recalculated. If this statement is not coded, the default is RRCI=2.

Example

RRCI=10

Related Topics

- The “SET RRCI” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SESAVG

The SESA VG statement specifies the estimated average number of NetEx sessions that is active for this host. NetEx uses this value to allocate buffer memory as needed for this host.

The SESA VG statement has the format:

Keyword (Select One)
<i>SESAVG=NUMBER-OF-SESSIONS</i> <i>SESAVG=32</i>

NUMBER-OF-SESSIONS

Specifies the average number of sessions (2 or greater). If this statement is not coded, the default is SESA VG=32.

Example

SESAVG=64

Related Topics

- “SESLIM” initialization statement on page 124
- “SESMAX” initialization statement on page 124

SESLIM

The SESLIM statement specifies the upper limit of the SESMAX parameter. Note that the SESMAX value is operator-changeable, but SESLIM is set at initialization and is an absolute limit.

The SESLIM statement has the format:

Keyword (Select One)
SESLIM=NUMBER-OF-SESSIONS
SESLIM=32

NUMBER-OF-SESSIONS

Specifies the maximum number of sessions (2 or greater). If this statement is not coded, the default is SESLIM=32.

Example

```
SESLIM=64
```

Related Topics

- “SESAVG” initialization statement on page 123
- “SESMAX” initialization statement on page 124

SESMAX

The SESMAX statement specifies the maximum number of concurrent Session connections supported by NetEx on this system. The count of connections includes both outstanding offers (SOFFER) and completed session connections. If a new session request is made when SESMAX sessions are in progress, the SOFFER or SCONNECT request is rejected with an error.

The SESMAX statement has the format:

Keyword (Select One)
SESMAX=NUMBER-OF-SESSIONS
SESMAX=32

NUMBER-OF-SESSIONS

Specifies the maximum number of sessions (from 2 to the SESLIM value). If this statement is not coded, the default is SESMAX=32.

Example

```
SESMAX=64
```

Related Topics

- The “SET SESMAX” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.
- “SESAVG” initialization statement on page 123
- “SESLIM” initialization statement on page 124

SMFTYPE

The SMFTYPE statement establishes the SMF record identifier number that NetEx uses when printing SMF records.

The SMFTYPE statement has the format:

Keyword (Select One)
SMFTYPE=RECORDTYPE
SMFTYPE=0

RECORDTYPE

Specify a value between 0 and 255. Note that record types 128-255 are reserved for user written records. The default is 00.

Example

```
SMFTYPE=16
```

Related Topics

See the following sections for more information on SMF record reporting:

- “SMF Record Layout” on page 17
- “IOCOUNT” initialization statement on page 96
- “XDISC” initialization statement on page 145

SMWTM

The SMWTM command specifies the amount of time (in seconds) that the NetEx session manager waits for a disconnect message after sending a confirm message. The default should be used unless otherwise recommended by Network Executive Software personnel.

The SMWTM statement has the format:

Keyword (Select One)
SMWTM=SECONDS
SMWTM=15

SECONDS

Specifies the session manager read for disconnect time in seconds. Allowed values are 1-60; the default is 15.

Example

```
SMWTM=20
```

Related Topics

- The “SET SMWTM” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SNDGRNM

The SNDGRNM specifies whether the group name or the real host name should be sent on connects. The use of this parameter will depend on the use of group hosts in the NCT in the site configuration.

The SNDGRNM statement has the format:

Keyword (Select One)
SNDGRNM=ON
SNDGRNM=OFF

The default is ON.

Example

SNDGRNM=OFF

SRBNAME

The SRBNAME statement specifies the SRB program name. A maximum of 8 characters are allowed.

The SRBNAME statement has the format:

Keyword (Select One)
SRBNAME= <i>SRB-NAME</i>
SRBNAME=NXMSRB01

SRB-NAME

Specifies the one to eight character SRB program name. If this statement is not coded, the default is SRBNAME=NXMSRB01.

Example

SRBNAME=NXMSRB02

SRCI

The SRCI statement specifies the send rate calculation interval (in seconds), used for protocol type-4 NetEx/IP connections. This value represents the time interval (in seconds) after which the send rate for a network connection is recalculated. The send rate refers to the rate at which the application is sending data from NetEx.

The SRCI statement has the format:

Keyword (Select One)
<i>SRCI=SEND-RATE-CALCULATION-INTERVAL</i> <i>SRCI=4</i>

SEND-RATE-CALCULATION-INTERVAL

Specifies the number of seconds (0-999) that must transpire, after which time the send rate for a network connection is recalculated. If this statement is not coded, the default is SRCI=4.

Example

SRCI=10

Related Topics

- The “SET SRCI” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SRPH

The SRPH statement specifies the send rate percentage high, used for protocol type-4 NetEx/IP connections. This value represents the high bound of the SRPS adjustment for each network connection. The value specified represents a percentage multiplied by a factor of 10.

The SRPH statement has the format:

Keyword (Select One)
<i>SRPH=SEND-RATE-PERCENTAGE-HIGH</i> <i>SRPH=500</i>

SEND-RATE-PERCENTAGE-HIGH

Specifies the high bound of the send and receive equivalence percentage. If this statement is not coded, the default is SRPH=500.

Example

SRPH=250

Related Topics

- “SET SRPH” command in H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual

SRPL

The SRPL statement specifies the send rate percentage low, used for protocol type-4 NetEx/IP connections. This value represents the low bound of the SRPS adjustment for each network connection. The value specified represents a percentage multiplied by a factor of 10.

The SRPL statement has the format:

Keyword (Select One)
SRPL= <i>SEND-RATE-PERCENTAGE-LOW</i> SRPL=100

SEND-RATE-PERCENTAGE-LOW

Specifies the low bound of the send and receive equivalence percentage. If this statement is not coded, the default is SRPL=100.

Example

SRPL=250

Related Topics

- The “SET SRPL” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SRPS

The SRPS statement specifies the send rate percentage equivalence, used for protocol type-4 NetEx/IP connections. This value represents the percentage factor used to increase or decrease the send rate for each network connection. The send rate may be increased after the expiration of the interval specified by RDII. The send rate may be decreased after the expiration of the interval specified by SRUI. The value specified represents a percentage multiplied by a factor of 10.

The SRPS statement has the format:

Keyword (Select One)
SRPS= <i>SEND-RATE-PERCENTAGE-EQUIVALENCE</i> SRPS=200

SEND-RATE-PERCENTAGE-EQUIVALENCE

Specifies the send and receive equivalence percentage. If this statement is not coded, the default is SRPS=200.

Example

SRPS=300

Related Topics

- The “SET SRPS” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SRUI

The SRUI statement specifies the send rate update interval (in seconds) , used for protocol type-4 NetEx/IP connections.

The SRUI statement has the format:

Keyword (Select One)
SRUI= <i>SEND-RATE-UPDATE-INTERVAL</i> SRUI=20

SEND-RATE-UPDATE-INTERVAL

Specifies the number of seconds (0-999) that must transpire, after which the send rate may be updated. If this statement is not coded, the default is SRUI=20.

Example

```
SRUI=30
```

Related Topics

- The “SET SRUI” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SSINAME

The SSINAME statement specifies the NetEx subsystem interface program name that resides in LPALIB. A maximum of 8 characters is allowed.

The SSINAME statement has the format:

Keyword (Select One)
SSINAME= <i>SUBSYSTEM-INTERFACE-NAME</i> SSINAME=NXMSSI

SUBSYSTEM-INTERFACE-NAME

Specifies the one to eight character NetEx subsystem interface program name. If this statement is not coded, the default is SSINAME=NXMSSI.

Example

```
SSINAME=ORTSSI
```

STRP

The STRP statement specifies the starting rate percentage, used for protocol type-4 NetEx/IP connections. This value represents a percentage factor used to calculate the initial send rate of each network connection. This value is applied against the maximum rate for the connection, as specified in the PAM, NRB, or by the DFRT parameter. During the course of the connection, the actual send rate may be adjusted, based on network activity. The value specified represents a percentage multiplied by a factor of 10.

The STRP statement has the format:

Keyword (Select One)
STRP= <i>STARTING-RATE-PERCENTAGE</i>
STRP=750

STARTING-RATE-PERCENTAGE

Specifies the starting rate percentage, applied against either the maximum rate for the connection, or against the DFRT parameter. If this statement is not coded, the default is STRP=750.

Example

```
STRP=500
```

Related Topics

- The “SET STRP” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

SUBSYS

The SUBSYS statement specifies the subsystem name used with this NetEx. A subsystem name is a 4-character entry that is placed in the z/OS Subsystem Name table. It identifies which remote address space an unauthorized application program wished to contact for cross-memory services. As a result, this parameter would normally be supplied when a “test” NetEx is being generated that runs concurrently with a “production” NetEx. To avoid conflicts with other subsystem names, the first 3 characters generated must be NET.

The SUBSYS statement has the format:

Keyword (Select One)
SUBSYS= <i>SUBSYSTEM-NAME</i>
SUBSYS=NETX

SUBSYSTEM-NAME

Specifies a 4-character subsystem name. If this statement is not coded, the default is SUBSYS=NETX.

Example

```
SUBSYS=NET1
```

SWTNAME

The SWTNAME statement specifies the SWAIT PC module name that resides in LPALIB. A maximum of 8 characters are allowed.

The SWTNAME statement has the format:

Keyword (Select One)
SWTNAME= <i>SWT-NAME</i>
SWTNAME=NXMSWTPC

SWT-NAME

Specifies the one to eight character SWAIT PC module name. If this statement is not coded, the default is SWTNAME=NXMSWTPC.

Example

SWTNAME=ORTSWT

TRACE

The TRACE statement specifies the trace options selected. The trace may be off, on memory or on tape, and on for either specified events or classes of events.

The TRACE statement has the format:

Keyword	Required Parameter (Select One)	Optional Parameters (Select One)	Optional Parameters (Select One)
TRACE=	ON	MEMORY MEMORY EVENTS MEMORY EVENTS=ALL MEMORY EVENTS=EVENTS TAPE=CCUU	CLASS CLASS=ALL CLASS=CLASS
	OFF		

ON

Enables the trace. Specify either ON or OFF.

OFF

Disables the trace. Specify either ON or OFF. If this statement is not coded, the default is TRACE=OFF.

MEMORY

Enables trace memory for the trace file. MEMORY EVENTS=ALL is the default for TRACE=ON.

EVENTS

(default between CLASS and EVENTS) This parameters specifies which trace events to save. You may select one or more trace events (shown in Table 5). If you do not select any events, the default is EVENTS=ALL.

TAPE=CUU

Sends trace to the tape unit specified by CUU.

CLASS

Specifies which class of events to save. You may select one or more trace classes (shown in Table 6). If you do not select any classes, the default is CLASS=ALL.

ALL

Saves all events.

CLASS

This parameter may be one or more trace classes.

Examples

```
TRACE ON MEMORY
TRACE ON MEMORY EVENTS=(DI,DO)
TRACE ON MEMORY EVENTS=(DI,DO) CLASS=MSG
TRACE ON TAPE=0290 CLASS=(CALL, MEM, MSG)
```

Table 5. Trace Events	
Trace Event	Description
<u>ALL</u>	All events listed in this table.
DI	Driver input
DO	Driver output
DS	Dispatch
ED	Enter dispatcher
IC	I/O completion
IH	Halt I/O
II	I/O start
MC	Module call
MF	Memory free
MG	Memory get
MR	Module return
QD	Queue dequeue
QQ	Queue queue
SP	Spawn
ST	Stammer
SU	Suspend
UR	User request in
US	User request out
XM	Cross-memory request

Table 6. Classes of Trace Events

Trace Class	Description
<u>ALL</u>	All classes listed in this table.
CALL	Call (Events MC, MR, SP, SU, ED, DS)
IO	I/O (Events DO, DI, II, IH, IC)
MEM	Memory (Events MG, MF)
MSG	Message (Events DO, DI)
QUE	Queue (Events QQ, QD)
TIME	Time (Event ST)
UREQ	User requests (Events UR, US)
XMEM	Cross-memory (Event XM)

Related Topics

- The “SET TRACE” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

TRCNO

The TRCNO statement specifies the number of buffers allocated for tracing purposes. The size of each trace buffer is specified by the TRCSIZE statement. Each of the buffers is filled in a circular fashion so that the oldest buffer is the one that is being filled. Sufficient buffers or buffer size should be allocated so that trace information is not lost by overflowing the trace output writing process.

The TRCNO statement has the format:

Keyword (Select One)
TRCNO= <i>NUMBER-OF-BUFFERS</i>
TRCNO=4

NUMBER-OF-BUFFERS

Specifies the number of trace buffers (0 or greater). If this statement is not coded, the default is TRCNO=4.

Example

TRCNO=8

Related Topics

- “TRCSIZE” initialization statement on page 135

TRCSIZE

The TRCSIZE statement specifies the size of each trace buffer in bytes. Trace records are accumulated chronologically in the trace buffer. When a buffer is full, the next trace buffer begins to fill. Sixteen bytes of the buffer are used as a header.

The TRCSIZE statement has the format:

Keyword (Select One)
TRCSIZE= <i>BYTES-PER-BUFFER-NAME</i>
TRCSIZE=8176

BYTES-PER-BUFFER

Specifies the number of bytes per buffer (1024 to 32760). If this statement is not coded, the default is TRCSIZE=8176.

Example

```
TRCSIZE=1024
```

Related Topics

- “TRCNO” initialization statement on page 134

TRMNAME

The TRMNAME statement specifies the Termination Resource Manager module name that resides in LPALIB. A maximum of 8 characters are allowed.

The TRMNAME statement has the format:

Keyword (Select One)
TRMNAME= <i>TERMINATIONNAME</i>
TRMNAME=NXMTERM0

TERMINATION-NAME

Specifies the one to eight character Termination Resource Manager module name. If this statement is not coded, the default is TRMNAME=NXMTERM0.

Note: Release 7.4 adds support that allows NetEx/IP to optionally be run in a reusable address space. The Release 7.4 version of the NXMTERMx module must be used with NetEx/IP Release 7.4, and is NOT compatible with previous versions. If it is necessary for NetEx/IP Release 7.4 to co-exist with previous versions of NetEx/IP in the same LPAR, then an alternate unique name must be specified for this module. This is done by specifying a different name on the NTXTERM parameter (e.g. NTXTERM(NXMTERMx) when running the NTXINST installation job. The name specified must then also be specified on the TRMNAME statement in the NetEx/IP NTXINIT startup parameters. This will result in the correct version of NXMTERMx to be loaded by this instance of NetEx/IP.

Example

```
TRMNAME=ORTTRM
```

TWOPART

The TWOPART statement specifies the length of time in milliseconds to delay between the first and second parts of a two-part message.

The TWOPART statement has the format:

Keyword (Select One)
TWOPART= <i>DELAY-TIME</i>
TWOPART=0

DELAY-TIME

Specifies the delay in milliseconds (0-9999). If this statement is not coded, the default is TWOPART=0.

Example

TWOPART=5

Related Topics

- The “SET TWOPART” command in the H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual.

UDPP

The UDPP statement specifies the UDP port number used by NetEx/IP. All NetEx/IP nodes in the network must use the same port number.

The UDPP statement has the format:

Keyword (Select One)
UDPP=UDP-PORTNUMBER
UDPP=6950

UDPP

Specifies the NetEx/IP UDP port number. The default is 6950.

Related Topics

None.

UIFNAME

The UIFNAME statement specifies the user interface program name that resides in LPALIB. A maximum of 8 characters is allowed.

The UIFNAME statement has the format:

Keyword (Select One)
UIFNAME= <i>INTERFACE-NAME</i>
UIFNAME=NXMUIF00

INTERFACE-NAME

Specifies the one to eight character user-interface name. If this statement is not coded, the default is UIFNAME=NXMUIF00.

Note: Release 7.4 adds support that allows NetEx/IP to optionally be run in a reusable address space. The Release 7.4 version of the NXMUIFxx module must be used with NetEx/IP Release 7.4, and is NOT compatible with previous versions. If it is necessary for NetEx/IP Release 7.4 to co-exist with previous versions of NetEx/IP in the same LPAR, then an alternate unique name must be specified for this module. This is done by specifying a different name on the NTXUIF parameter (e.g. NTXUIF(NXMUIFxx) when running the NTXINST installation job. The name specified must then also be specified on the UIFNAME statement in the NetEx/IP NTXINIT startup parameters. This will result in the correct version of NXMUIFxx to be loaded by this instance of NetEx/IP.

Example

UIFNAME=ORTUIF

USERNAME

The USERNAME statement specifies the name of the user PC module that resides in LPALIB. A maximum of eight characters is allowed.

The USERNAME statement has the format:

Keyword (Select One)
USERNAME= <i>MODULE-NAME</i>
USERNAME=NXMUSRPC

MODULE-NAME

Specifies the one to eight character name of the user PC module. If this statement is not coded, the default is USERNAME=NXMUSRPC.

Example

USERNAME=ORTUSER

USERPK

The USERPK statement specifies the program key(s) that authorize an application program to invoke the calls to Netex. This statement is only necessary when Netex application programs are set up to run in a program key other than 8. If this statement is used, the specified key(s) are allowed in addition to key 8.

Keyword (Select One)
USERPK= <i>key</i>
USERPK=(<i>key1</i> , <i>key2</i> , <i>key3</i> , , , . <i>keyn</i>)
USERPK=8

KEY

Specifies a value in the range of 1-15, and may consist of a single key or a list of keys. If this statement is not coded, a value of USERPK=8 is assumed, which is the normal program key used for z/OS problem state programs.

Example

```
USERPK= ( 3 , 4 )  
USERPK=6
```

Note: USERPK can NOT be set with a corresponding operator command. If it is necessary to set USERPK keys, they can only be specified during Netex initialization.

USERSK

The USERSK statement specifies the storage key(s) that are used by Netex application programs. This statement is only necessary when Netex application programs are set up to run in a program key other than key 8, and use storage for Netex data buffers in a key other than the specified program key. If this statement is used, the specified key(s) are allowed in addition to key 8 and the keys specified by USERPK.

Keyword (Select One)
USERSK= <i>key</i>
USERSK=(<i>key1</i> , <i>key2</i> , <i>key3</i> , , , . <i>keyn</i>)
USERSK=8

KEY

Specifies a value in the range of 1-15, and may consist of a single key or a list of keys. If this statement is not coded, a value of USERSK=8 is assumed, which is the normal storage key used for z/OS problem state programs.

Example

```
USERSK= ( 5 , 6 )  
USERSK=7
```

Note: USERSK can NOT be set with a corresponding operator command. If it is necessary to set USERSK keys, they can only be specified with this initialization statement.

WDOGIN T

The WDOGIN T statement specifies the number of seconds elapsed between events timed by the Low Resolution timer. Many internal tasks in NetEx use this timer to allow the work in progress to abort if substantial delays occur. The CONTIME, IDLETIME, and READTIME parameters (described above) uses this timer. Every WDOGIN T seconds, the Low Resolution timer code is activated and decrements the elapsed time available for all internal tasks. Any time expiration such as CONTIME must therefore actually take place after an interval that is a multiple of WDOGIN T.

The WDOGIN T statement has the format:

Keyword (Select One)
WDOGIN T=SECONDS
WDOGIN T=2

SECONDS

Specifies the number of seconds (0-999) between events. If this statement is not coded, the default is WDOGIN T=2.

Example

WDOGIN T=4

Related Topics

- “SET WDOGIN T” command in H210IPZ NetEx/IP for IBM z/OS Operating Systems Operator Reference Manual

XCONF

The XCONF statement activates or disables the user “confirm” exit. Refer to “User Exits” on page 149 for more information.

The XCONF statement has the format:

Keyword (Select One)
XCONF=ON
XCONF=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XCONF=OFF.

Example

```
XCONF=ON
```

Related Topics

- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XCONN

The XCONN statement activates or disables the user “connect” exit. Refer to “User Exits” on page 149 for more information.

The XCONN statement has the format:

Keyword (Select One)
XCONN=ON
XCONN=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XCONN=OFF.

Example

```
XCONN=ON
```

Related Topics

- “XCONF” initialization statement on page 140
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XCXN

The XCXN statement activates or disables the user “connection” exit. Refer to “User Exits” on page 149 for more information.

The XCXN statement has the format:

Keyword (Select One)
XCXN=ON
XCXN=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XCXN=OFF.

Example

```
XCXN=ON
```

Related Topics

- “XCONF” initialization statement on page 149
- “XCONN” initialization statement on page 141
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “User Exits” initialization statement on page 149
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148

XDATAI

The XDATAI statement activates or disables the user data input exit. Refer to “User Exits” on page 149 for more information.

The XDATAI statement has the format:

Keyword (Select One)
XDATAI=ON
XDATAI=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XDATAI=OFF.

Example

```
XDATAI=ON
```

Related Topics

- “XCONF” initialization statement on page 140
- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XDATAO

The XDATAO statement activates or disables the user “data output” exit. Refer to “User Exits” on page 149 for more information.

The XDATAO statement has the format:

Keyword (Select One)
XDATAO=ON
XDATAO=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XDATAO=OFF.

Example

```
XDATAO=ON
```

Related Topics

- “XCONF” initialization statement on page 140
- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDISC” initialization statement on page 145
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XDISC

The XDISC statement activates or disables the user “disconnection” exit. Refer to “User Exits” on page 149 for more information.

The XDISC statement has the format:

Keyword (Select One)
XDISC=ON
XDISC=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XDISC=OFF.

Example

```
XDISC=ON
```

Related Topics

- “SMF Record Layout” on page 17
- “IOCOUNT” initialization statement on page 96
- “SMFTYPE” initialization statement on page 125
- “XCONF” initialization statement on page 140
- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XINIT

The XINIT statement activates or disables the user “initialization” exit. Refer to “User Exits” on page 149 for more information.

The XINIT statement has the format:

Keyword (Select One)
XINIT=ON
XINIT=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XINIT=OFF.

Example

```
XINIT=ON
```

Related Topics

- “XCONF” initialization statement on page 140
- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “XOFFR” initialization statement on page 147
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XOFFR

The XOFFR statement activates or disables the user offer exit. Refer to “User Exits” on page 149 for more information.

The XOFFR statement has the format:

Keyword (Select One)
XOFFR=ON
XOFFR=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XOFFR=OFF.

Example

```
XINIT=ON
```

Related Topics

- “XCONF” initialization statement on page 140
- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “XINIT” initialization statement on page 146
- “XTERM” initialization statement on page 148
- “User Exits” on page 149

XTERM

The XTERM statement activates or disables the user termination exit. Refer to “User Exits” on page 149 for more information.

The XTERM statement has the format:

Keyword (Select One)
XTERM=ON
XTERM=OFF

ON

Activates the exit.

OFF

Disables the exit. If this statement is not coded, the default is XTERM=OFF.

Example

```
XTERM=ON
```

Related Topics

- “XCONF” initialization statement on page 140
- “XCONN” initialization statement on page 141
- “XCXN” initialization statement on page 142
- “XDATAI” initialization statement on page 143
- “XDATAO” initialization statement on page 144
- “XDISC” initialization statement on page 145
- “XINIT” initialization statement on page 146
- “XOFFR” initialization statement on page 147
- “User Exits” on page 149

User Exits

General

NetEx provides user exits to allow installations to maintain additional control over the connection of NetEx applications and the movement of data between them. Since the exits become part of the NetEx load module, code inserted in them has complete addressability to everything within NetEx.

CAUTION

Be careful when adding or changing code in the exits and follow strict control procedures when testing them.

A set of dummy exits are distributed with NetEx that become part of the NetEx load module. To use a particular exit, the customer must insert the code into the appropriate exit, assemble it and relink it with the NetEx load module. A flag byte in the ROOT control block defines which exits to take.

When an exit is activated, it is activated for every application that uses NetEx. An application cannot selectively activate or deactivate exits. This is not really a restriction, since the user's exit can provide checking or bypassing.

The rest of this section describes the following topics:

- General Exit Interface
- Specific Exits
- Exit Installation
- Programming Considerations (including examples)
- Miscellaneous Considerations
- Examples

General Exit Interface

Exit points are located such that they can be taken before, during, or after NetEx connections are established. Additional exits can be taken at NetEx initialization and termination.

All of the connection user exits are entered from the NetEx Request Director. The initialization and termination exits are entered from the NetEx initialization module.

The NetEx NTCCALL macro invokes each exit. The exit saves and restores registers through the NetEx NTCENTER and NTCLEAVE macros. These macros are coded in the distributed dummy exits.

If any connection exits are activated, NetEx obtains a 100-byte connection work area before the first exit is issued for a specific connection. This storage is initialized to binary zeroes before presenting it to the first exit and remains available for all subsequent exits to use that are taken during that connection. Data stored in this area by an exit remains intact until the connection is terminated.

NetEx maintains a connection statistics block that the disconnection exit can access. This block contains statistics for the connection, such as numbers of reads and writes, and numbers of re-transmissions.

Additionally, you can use a local 64 byte work area located at SAVEWORK (addressable from register 13) as a work area for this particular exit execution.

If any other storage is obtained in the exit (either using an operating system request, or a NetEx storage request), the user exit must free the storage when finished. The user connection work area is freed automatically by NetEx when the connection terminates. The user exit should never free this storage.

A user field is defined in the ROOT control block (label ROOTUSER) which is available for the exits to use. Since the ROOT is always addressable from the exits, this user field is always accessible. A typical use of this field might be to store the address of some global storage, obtained during the initialization exit that would be accessible from all of the other exits.

NRB status codes returned by an exit must be in the range of 9000-9099 for a local request and 9100-9199 for a remote request. Any invalid return code or status code is replaced by the default. Parameters returned from an exit must be in register format.

The parameters which NetEx passes to the user exits can no longer be wholly contained in registers 1-4. A parameter list is used instead (except with UX1); the actual contents of the fields vary depending on the exit invoked. In most cases, the end of the parameter list is flagged with the highest-order bit set to 1. Table 7 shows the format of the parameter list.

Table 7. Parameter List Fields			
R1 Points to STAUCXWA Entry to All User Exits			
STAUCXWA	DS	A	User Work Area Address
STNRBADR	DS	A	NetEx NRB Address
STPDAXX	DS	A	PDATA Address
Contents Contingent Upon Exit			
STPDLENG	DS	F	PDATA Length
STODADDR	DS	A	ODATA Address
STODLENG	DS	F	ODATA Length
STPRMLEN	EQU	*-STAUCXWA	PARM Section Length
SCXWASIZ	EQU	*-CXWA	Size of system section
User Connected Work Area			
UCXWA	DS	0F	
UCWASIZ	EQU	100	Size of Connected Work Area

Specific Exits

Each exit is provided as an entry in an exit module. The exit module names are NXMUXn, where n is an identifying digit. Table 8 identifies the individual exit module names and entry points. Table 7 lists the parameter list fields used by the exit modules. Contents of the fields are given in the description of each exit module.

Table 8. User Exit Module Names and Entry Points		
Module	Entry	Function
NXMUX1	INNXXMUX1IN	User NetEx initialization
	NXXMUX1TM	User NetEx termination
NXMUX2	NXXMUX2OF	User "offer" exit
	NXXMUX2CN	User "connect" exit
	NXXMUX2CF	User "confirm" exit
NXXMUX3	NXXMUX3CX	User "connection" exit
NXMUX4	NXXMUX4DI	User "data in" exit
	NXXMUX4DO	User "data out" exit
NXXMUX5	NXXMUX5DC	User "disconnect" exit. Currently used in the SMF record as the generation exit. For more information on SMF records, see "SMF Considerations" on page 203.

Each exit listed in Table 8 is described in the following paragraphs.

NXMUX1IN

This exit is entered during NetEx initialization.

This exit provides support for user initialization processing. During this exit, the user may perform some global initialization that may be necessary for subsequent connection exits. For example, a request for additional NetEx storage could be made from this exit. This storage could be used for such things as a user DCB, a user validation table, accounting numbers, or anything else the user requires. If this storage address is saved in the ROOT control block (at ROOTUSER), this data would then be available for subsequent connection exits to use.

Entry Specification

The entry specification for this exit is shown below:

UX1IN NTCENTER

Entry Conditions

The following registers are available on entry:

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area

R14

return address

R15

entry address before NTCENTER

Exit Specification

The exit specification for this exit is shown below:

[label] NTCLEAVE

Exit Conditions

There are no exit conditions for this exit.

NXMUX1TM

The NXMUX1TM exit is entered during NetEx termination.

This exit provides support for user termination processing. During this exit, the user may perform some global termination processing, such as closing a file in which connection exits were recorded. It is not necessary to free previously acquired NetEx storage in this exit, since NetEx is terminating.

Entry Specification

The entry specification for this exit is shown below:

```
UX1TM      NTCENTER NOSAVE
```

Entry Conditions

The following registers are available on entry:

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER NOSAVE: address of save area for NTCENTER to use following NTCENTER
NOSAVE: same

R14

return address

R15

entry address before NTCENTER

Exit Specification

The exit specification for this exit is shown below:

```
[label]    NTCLEAVE
```

Exit Conditions

There are no exit conditions for this exit.

Restrictions

NOSAVE must be specified on entry to this exit. Since this exit is entered at NetEx termination, no assumptions can be made concerning the availability of a new save area. Consequently, requests for any other NetEx services cannot be made from this exit.

NXMUX2OF

The NetEx Request Director issues this exit before it issues an offer request.

This exit provides the installation with some control over NetEx use. Individual OFFER requests can be validated and/or modified as desired, and subsequently they can be accepted or rejected.

Entry Specification

The entry specification for this exit is shown below:

UX2OF NTCENTER

Entry Conditions

The following list describes the contents of the parameter list fields upon entry:

STAUCXWA

user's 100-byte workarea address

STNRBADR

highest order bit is "1" (EOL) NRB address

STPDAXX

(unused; 0)

STPDLENG

(unused; 0)

STODADDR

(unused; 0)

STODLENG

(unused; 0)

The following registers are available on entry:

R1

address of connection work area parameter list

R2

address of the user's NRB

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area

R14

return address

R15

entry address before NTCENTER

The following fields identify the requestor:

NRBPNAME

name that the user wishes to offer

NITUSER

jobname of offering user

ROOTHOST

name of offering (local) host

The following fields in the NRB may be analyzed and/or modified:

NRBDMODE

assembly/disassemble mode

NRBTIME

offer timeout value

NRBCLASS

class of service desired

NRBMAXRT

maximum rate of transmission

NRBBLKI

block size to use for input

NRBBLKO

block size to use for output

NRBPNAME

name that is offered

Exit Specification

The exit specification for this exit is shown below:

```
[label]    NTCLEAVE,[param1],[param2]
```

Exit Conditions

The following return codes can be returned in “param1:”

0

accept request

12

reject request, return 9000 in NRBSTAT

16

reject request, store “param2” in NRBSTAT

NXMUX2CN

The NetEx Request Director issues this exit before it issues a connect request.

This exit provides the installation with some control over NetEx use. Individual CONNECT requests can be validated and/or modified as desired, and subsequently they can be accepted or rejected.

Entry Specification

The entry specification for this exit is shown below:

UX2CN NTCENTER

Entry Conditions

The following list describes the contents of the parameter list fields upon entry:

STAUCXWA

user's 100-byte workarea address

STNRBADR

NRB address

STPDAXX

PDATA address

STPDLENG

data length

STODADDR

CLASS 0,2

ODATA address

CLASS 1

always 0

STODLENG

highest order bit is "1" (EOL)

CLASS 0,2

ODATA length

CLASS 1

always 0

The following registers are available on entry:

R1

address of connection work area parameter list

R2

address of the user's NRB

R3

address of user PDATA

R4

length of user PDATA

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area

R14

return address

R15

entry address before NTCENTER

The following fields identify the requestor:

NITUSER

jobname of connecting user

ROOTHOST

name of connecting (local) host

NRBPNAME

offered name to which connection is being attempted

NRBHNAME

name of host to whom connection is being attempted

The following fields in the NRB may be analyzed and/or modified:

NRBDMODE

assembly/disassemble mode

NRBCLASS

class of service desired

NRBMAXRT

maximum rate of transmission

NRBBLKI

block size to use for input

NRBBLKO

block size to use for output

Exit Specification

The exit specification for this exit is shown below:

[label] NTCLEAVE, [param1], [param2]

Exit Conditions

The following return codes can be returned in “param1:”

0

accept request

- 12** reject request, return 9010 in NRBSTAT
- 16** reject request, store “param2” in NRBSTAT

NXMUX2CF

The NetEx Request Director issues this exit before it issues a confirm request.

This exit provides the installation with some control over NetEx use. Individual CONFIRM requests can be validated and/or modified as desired, and subsequently they can be accepted, rejected, or disconnected.

Entry Specification

The entry specification for this exit is shown below:

UX2CF NTCENTER

Entry Conditions

The following list describes the contents of the parameter list fields upon entry:

STAUCXWA

user's 100-byte workarea address

STNRBADR

NRB address

STPDAXX

PDATA address

STPDLENG

PDATA length

STODADDR

CLASS 0,2

ODATA address

CLASS 1

always 0

STODLENG

highest order bit is "1" (EOL)

CLASS 0,2

ODATA length

CLASS 1

always 0

The following registers are available on entry:

R1

address of connection work area parameter list

R2

address of the user's NRB

R3

address of user PDATA

R4

length of user PDATA

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER: address of save area for NTCENTER to use following NTCENTER: of new save area

R14

return address

R15

entry address before NTCENTER

The following fields identify the requestor:

NITUSER

virtual machine ID of confirming user

ROOTHOST

name of confirming host

NRBHNAME

name of remote host connected to

NRBPNAME

OFFER name connected to

The following fields in the NRB may be analyzed and/or modified:

NRBLEN

length of data to write (same as reg 4)

NRBDMODE

assembly/disassembly mode

NRBBLKO

negotiated block size to use for output

Exit Specification

The exit specification for this exit is shown below:

```
[label]    NTCLEAVE, [param1], [param2]
```

Exit Conditions

The following return codes can be returned in “param1:”

0

accept request

12

reject request and connection, return 9030 in NRBSTAT, disconnect the remote with 9130

16

reject request and connection, store “param2” in NRBSTAT disconnect the remote with “param3”

NXMUX3CX

The NetEx Request Director issues this exit after a connection is fully established (either an OFFER or a READ CONFIRM completes successfully).

This exit provides the installation with some control over NetEx use. Individual connections can be validated as desired, and subsequently they can be accepted or disconnected. During this exit, the user may create a connection record in the 100 byte connection work area. This record can then be written to a file either from this exit or from a subsequent exit for this connection.

Entry Specification

The entry specification for this exit is shown below:

UX3CX NTCENTER

Entry Conditions

The following list describes the contents of the parameter list fields upon entry:

STAUCXWA

user's 100-byte workarea address

STNRBADR

NRB address

STPDAXX

SOFFER completion connection

highest order bit set to "1" (EOL)

address of two contiguous eight-byte fields containing:

- connecting host name
- connecting user name

SCONNECT completion connection

- negative 1 (X'FFFFFFFF')

STPDLENG

(unused; 0)

STODADDR

(unused; 0)

STODLENG

(unused; 0)

The following registers are available on entry:

R1

address of connection work area parameter list

R2

address of the user's NRB

R3

if OFFER completed: address of 16 byte field containing eight-byte connecting host name, followed by eight-byte connecting jobname or user ID.

If CONNECT completed: negative number

R4

CXWA address

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area

R14

return address

R15

entry address before NTCENTER

The following fields identify the local and remote users.

If OFFER completes:

NRBPNAME

name of connecting user

NITUSER

job name of offering user

ROOTHOST

name of offering host

Reg 3

address of 16 byte field containing the eight-byte connecting host name, followed by eight-byte connecting user name. The nature of the user jobname or ID depends on the type and operating system being used.

If READ CONFIRM completes:

NITUSER

name of connecting user

ROOTHOST

name of connecting host

NRBPNAME

offered name of user connected to

NRBHNAME

name of host connected to

The following fields in the NRB may also be analyzed:

NRBDMODE

assembly/disassemble mode

NRBCLASS

class of service established

NRBMAXRT

maximum rate of transmission

NRBBLKI

negotiated block size to use for input

NRBBLKO

negotiated block size to use for output

Exit Specification

The exit specification for this exit is shown below:

```
[label] NTCLEAVE,[param1],[param2],[param3]
```

Exit Conditions

The following return codes can be returned in “param1:”

0

accept connection

4

if offer completion, disconnect remote with 9120, reissue offer request

8

if offer completion, disconnect remote with “param3”, reissue offer request

12

reject connection, return 9020 in NRBSTAT, disconnect remote with 9120

16

reject connection, store “param2” in NRBSTAT, disconnect remote with “param3”

NOTE: If NetEx reissues an Offer request following rejection of a connection, the timeout value for the re-issued Offer is the time remaining from the original request. The offering application will think that the connection has never taken place. In this case, neither the disconnect exit nor the offer exit is re-entered on the host re-issuing Offer.

NXMUX4DI

The NetEx Request Director issues this exit before presenting user data to a local application.

This exit provides the installation with some control over NetEx use. The amount or type of data can be analyzed and/or modified in the exit. During this exit, the user may build a data record or add information to a previously created record. This record can be built or maintained in the 100 byte connection work area, and can be written to a file either from this exit or from a subsequent exit for this connection.

Entry Specification

The entry specification for this exit is shown below:

UX4DI NTCENTER

Entry Conditions

The following list describes the contents of the parameter list fields upon entry:

STAUCXWA

user's 100-byte workarea address

STNRBADR

NRB address

STPDAXX

CLASS 0,2

block descriptor address

CLASS 1

PDATA address

STPDLENG

PDATA length

STODADDR

CLASS 0,2

ODATA address

CLASS 1

always 0

STODLENG

highest order bit is "1" (EOL)

CLASS 0,2

ODATA length

CLASS 1

always 0

NOTE: Class 0,2 PDATA consists of a variable number of segments anchored by a block descriptor (that is, the user's PDATA at this point is not contained in one integral unit). The address passed in STPDAXX is the address of the data block descriptor.

Refer to the NTCBLKDS macro for the layout of the data block descriptor, and to the NTCSEGH macro for the layout of the segment header.

The following registers are available on entry:

- R1**
address of connection work area parameter list
- R2**
address of the user's NRB
- R3**
address of user PDATA
- R4**
length of user PDATA
- R8**
address of NetEx ROOT control block
- R9**
address of current NIT
- R12**
base address following NTCENTER
- R13**
before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area
- R14**
return address
- R15**
entry address before NTCENTER

The following fields identifies the user:

- NITUSER**
jobname
- NRBPNAME**
offered name of user connected to
- NRBHNAME**
name of host connected to

The following fields in the NRB may be useful in this exit:

- NRBLEN**
length of data read (same as reg 4)
- NRBUBIT**
unused bit count
- NRBDMODE**
assembly/disassembly mode

NRBBLKI

negotiated block size to use for input

Exit Specification

The exit specification for this exit is shown below:

[label] NTCLEAVE,[param1],[param2]

Exit Conditions

The following return codes can be returned in “param1:”

0

accept data

12

do not present data to user, return 9050 in NRBSTAT

16

do not present data to user, store “param2” in NRBSTAT

NXMUX4DO

The NetEx Request Director issues this exit after it accepts user data from a local application, but before sending it to the remote application.

This exit provides the installation with some control over NetEx use. The amount or type of data can be analyzed and/or modified in the exit. During this exit, the user may build a data record, or add information to a previously created record. This record can be built or maintained in the 100 byte connection work area, and can be written to a file either from this exit or from a subsequent exit for this connection.

Entry Specification

The entry specification for this exit is shown below:

UX4DO NTCENTER

Entry Conditions

The following list describes the contents of the parameter list fields upon entry:

STAUCXWA

user's 100-byte workarea address

STNRBADR

NRB address

STPDAXX

PDATA address

STPDLENG

PDATA length

STODADDR

CLASS 0,2

ODATA address

CLASS 1

always 0

STODLENG

highest order bit is "1" (EOL)

CLASS 0,2

ODATA length

CLASS 1

always 0

The following registers are available on entry:

R1

address of connection work area parameter list

R2

address of the user's NRB

R3

address of user PDATA

- R4**
length of user PDATA
- R8**
address of NetEx ROOT control block
- R9**
address of current NIT
- R12**
base address following NTCENTER
- R13**
before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area
- R14**
return address
- R15**
entry address before NTCENTER

The following fields identifies the user:

NITUSER

job name

NRBPNAME

offered name of user connected to

NRBHNAME

name of host connected to

The following fields in the NRB may be useful in this exit:

NRBLEN

length of data to write (same as reg 4)

NRBDMODE

assembly/disassembly mode

NRBBLKO

negotiated block size to use for output

Exit Specification

The exit specification for this exit is shown below:

[label] NTCLEAVE, [param1], [param2]

Exit Conditions

The following return codes can be returned in “param1:”

- 0**
accept data
- 12**
reject data, return 9060 in NRBSTAT

16

reject data, store “param2” in NRBSTAT

STAUCXWA

user's 100-byte workarea address

STNRBADR

NRB address

STPDAXX

highest order bit is "1" (EOL) CXWA address

STPDLENG

(unused)

STODADDR

(unused)

STODLENG

(unused)

The following registers are available on entry:

R1

address of connection work area parameter list

R2

address of an NRB representing an internally generated disconnect

R3

address of connection statistics block

R8

address of NetEx ROOT control block

R9

address of current NIT

R12

base address following NTCENTER

R13

before NTCENTER: address of save area for NTCENTER to use following NTCENTER: address of new save area

R14

return address

R15

entry address before NTCENTER

The following field identifies the user:

NITUSER

job name

Exit Specification

The exit specification for this exit is shown below:

[label] NTCLEAVE

Exit Conditions

Connection work area is no longer available.

Exit Installation

To implement user exits, the user must change the provided dummy exits to include any additional processing. The exits must then be assembled and linked with the NetEx load module, using the NTXLINK sample JCL provided on file 6 on the installation tape.

Before the exits are activated, the corresponding exit parameters must be specified either on the NTCROOTG macro, or in the initialization file. See “Initialization Statements” on page 73 for a description of these parameters.

Programming Considerations

The following subsections describe the topics listed below:

- Standards and Conventions
- Available Services
- Miscellaneous Considerations
- Examples

Standards and Conventions

The following paragraphs describe programming standards developed for the NetEx modules. These standards must be applied to the user exits. These standards are listed below:

- NTCOPT Macro
- NTCMOD Macro
- NTCCALL Macro
- NTCENTER Macro
- NTCLEAVE Macro
- Copy Members

NTCOPT Macro

Copy member NTCOPT contains global SETC declarations required by other macros. It must be the first statement in the exit, and is provided in the distributed dummy exits.

NTCMOD Macro

The NTCMOD macro begins the assembly of a NetEx module. It identifies the operating system (MVS) on which this NetEx is executing. It generates a CSECT name for the module, which consists of a specified name prefixed by a three character operating system identification (NXM for MVS). It establishes addressability for the ROOT (register 8), the NIT (register 9), and the SAVEAREA (register 13). It generates an eyeball that consists of the 8-character module name, the date of assembly, the release and service level, and the SPLEVEL value at the time of assembly.

The format of the NTCMOD macro request is shown below:

Label	Macro	Required Positional Parameter	Optional Keyword Parameters (Select appropriate parameters)
<i>LABEL</i>	NTCMOD	MVS	A= <u>24</u> A= <u>31</u> A=ANY R= <u>24</u> R=ANY NETEX= <u>YES</u> NETEX=NO

LABEL

This is a one to five character name that generates the CSECT name of the module (prefixed by NXM).

MVS

Specifies the host operating system (MVS).

A

Specifies the addressing mode.

R

Specifies the residency mode.

NETEX

Specifies whether a “USING” assembler instruction is generated for the NIT, ROOT, and SAVEAREA control mapping block dsect.

YES

Generates the “USING” assembler instruction.

NO

Disables the “USING” assembler instruction.

The NTCMOD generates the code in Figure 45 for a z/OS system:

```

csect      NTCMOD MVS                      SPECIFY OS;
          FIX=QX42559                      SPECIFY LATEST FIX APPLIED
*****
*          GLOBAL SYMBOLIC VARIABLE &SYSSPLV
*          HAS A VALUE OF: 4
*****
NXMcsect   CSECT ,                        NAMED CSECT
          AMODE 31                        SET MVS/VM XA/ESA AMODE
          RMODE 24                        SET MVS/VM XA/ESA RMODE
          BASR  R15,R0                    A(NSI) INTO &REG
          B      @@A0001C-*(,R15)         GO IF NO 'ENTRY' DECLARED
          DC     AL01(@A0001C-(*+01)      L'I-BALLS NOT THIS FIELD
          DC     CL08'NXMADA'              CSECT NAME
          DC     CL08'04/11/97'            DATE ASSEMBLED
          DC     CL08'12.36'              TIME ASSEMBLED
          DC     CL08'MVS/ESA'             OPERATING SYSTEM MNEMONIC
          DC     CL08'SPLEVEL4'            SPLEVEL USED 4 ASSEMBLE
          DC     CL08'03029702'            RELEASE AND SERVICE LVL
          DC     CL08'QZ42559'            LATEST FIX APPLIED
          DC     C' COPYRIGHT 1997,'        OFFICIAL
          DC     C' STORAGE TECHNOLOGY CORPORATION,' COPYRIGHT
          DC     C' ALL RIGHTS RESERVED.   ' STATEMENT
          DC     000D                      D/W PAD-OUT THE LITERALS

```

Figure 45. NTCMOD for a z/OS System

NTCCALL Macro

The NTCCALL macro performs a synchronous call to another module of NetEx. When the called module receives control, it commences by using the NTCENTER macro to save the caller's registers and establish addressability. When the called module has finished its function, it executes the NTCLEAVE macro which frees the register save area acquired during the entry to the module and returns control to the point following the NTCCALL macro in the calling module. The format of the NTCCALL macro request is shown below:

Label	Macro	Required Parameter	Optional Parameters
<i>LABEL</i>	NTCCALL	<i>NAME</i>	<i>, PARAM1 , PARAM2 , PARAM3 , PARAM4</i>

LABEL

This is generated at the first executable address in the macro code.

NAME

Specifies the name of the module called, minus the 3 character operating system prefix. It must be a CSECT or ENTRY point elsewhere in NetEx.

PARAM1 through PARAM4

This optional parameter is as many parameters may be included as are needed by the called module. When the module is called, registers 1 through 4 respectively contain the values specified in PARAM1 through PARAM4. If a parameter value is omitted, the contents of the respective register are not modified. Parameters may have the form (register) or address. The address must be RX instruction addressable.

If tracing is in effect, NTCCALL first goes to the Module Call entry of the trace routine. This entry records the entry into the module and the parameters passed to it.

Upon entry to the called module, R15 contains the address of the entry point in the called module, R14 contains the return address in the calling module.

The called module returns control to the point following the NTCCALL macro. If tracing is in effect, the NTCCALL macro generates a call to the Module Return entry of the trace routine, which records the fact that control is returned to the original calling module.

Each user exit is called (NTCCALL) from the appropriate exit point in NetEx. The user exit routines must save registers upon entry and restore them when returning. If any other NetEx services or operating system services are invoked from the exit, then another save area must be provided by the exit. This save area must be freed upon leaving the exit. The NetEx macros NTCENTER and NTCLEAVE accomplishes this task. When using NTCLEAVE, up to four registers (registers 1-4) can be used for returning values. The macro generates the following code:

label	NTCCALL	name	
	L	0	=V(NXMname)
	L	R15	=V(NXMTRCMC)
	BALR	R14,R15	ISSUE MODULE CALL TRACE
	L	15	=V(NXMname)
	BALR	R14,R15	GO THERE
	L	R15	=V(NXMTRCMR)
	BALR	R14,R15	
*			

NTCENTER Macro

The NTCENTER macro executes whenever a user exit receives control from a calling module. It establishes addressability to R12 at the beginning of the module, and it saves the caller's registers into the current save area. The NTCENTER request has the following format.

Label	Macro	Optional Parameter
<i>LABEL</i>	NTCENTER	NOSAVE

LABEL

Specifies the name of the entry point generated as part of the NTCENTER macro. The label specified should consist of a 1-5 character name and must be different from that specified on the NTCMOD macro. The label specified is prefixed with the three character operating system identification (NXM for MVS). The macro generates an ENTRY label statement and places the label at the start of the code generated by the macro.

NOSAVE

Specifies if the NTCENTER macro is to acquire a new save area or not. When NTCENTER is first executed, it saves the registers on entry into the save area provided by the calling module. If NOSAVE is omitted, NTCENTER next calls the Memory Manager to obtain another save area. This second save area is chained to the first, and R13 points to the new save area upon completion of the NTCENTER macro. The size of this newly obtained save area is SAVESIZE. SAVESIZE is a system parameter that contains sufficient room to save all registers plus 64 bytes of working storage which can be used as a work area by the exit.

The code shown in Figure 46 is generated by NTCENTER for a z/OS system:

label	NTCENTER ,	
	ENTRY NXMlabel	DECLARE EP
	CNOP 0,8	F/W ALIGN EP IBALL
	DC CL08'NXMlabel'	ENTRY EYEBALL
NXMlabel	DS 000H	ENSURE H/W ALIGNMENT
	STM R14,R12,SAVER14	STANDARD SAVE
	BALR R12,R0	NSI INTO 1ST BASE
	USING *,R12	TEMP PROG BASE *
	L R12,E0004	CSECT ADDR
	BSM R12,R0	GET AMODE BIT INTO BASE REG
	DROP R12	RELEASE INITIAL BASE REG
	USING NXMcsect,R12	
	B N0004	B AROUND CSECT ADCON
	CNOP 0,4	ALIGN FOR ADCON
E0004	DC A(NXMcsect)	BASE ADDRESS LABEL
N0004	DS 00H	TAG POINT
	LM R14,15,ROOTSAFR	
MEM0005A	DS 000H	
	LTR R14,R14	A(SAVE-AREA): +?
	BNZ MEM0005B	NO: BRANCH, BLOCKS AVAILABLE
	DS 000H	TAG POINT
	L R15,=V(NXMINIAB)	A(ABEND NETEX EP)
	BASR R0,R15	GO
	DC AL1(255),CL3'csect'	CODE,MODULE-ID
MEM0005B	DS 000H	BLOCK LOCATED
	L R0,0000(,R14)	A(FOLLOWING SAVE-AREA)
	LR R1,R15	COPY CURRENT COUNT
	AL R1,=F'001"	INCREMENT COUNT
	CDS R14,R0,ROOTSAFR	EXTRACTED OK?
	BNE MEM0005A	
	LR R0,R14	COPY ASAVE-AREA EXTRACTED)
	L R1,=A(SAVESIZE)	L'SAVE-AREA
	SLR R15,R15	
	MVCL R0,R14	CLEAR MEMORY
	LR R15,R14	
	ST R13,SAVEBACK-SAVEAREA(R15)	BACK CHAIN
	ST R15,SAVENEXT	FORWARD CHAIN TO NEW
	LM R13,R1,SAVENEXT	RESTORE R13-R1
	ST R13,NITSAVE	A(NEW SAVE-AREA) IN NIT
	MVC SAVERESV,=CL004'ADAI'	

Figure 46. NTCENTER Macro for a z/OS System

Upon completion of the NTCENTER macro, R12 points to the origin of the module, R14 points to the calling module's return address but is available for other use, and R5 through R11 are the same as in the calling module, and R1 through R4 contain the parameters specified by the caller's NTCCALL macro.

NTCLEAVE Macro

The user exit uses the NTCLEAVE macro to return control to a calling module. It extracts the calling module's return address from the save area, calls the Memory Manager to free the save area acquired when the module was first entered, makes the calling module's save area once again current, and branches back to the calling module.

Label	Macro	Optional Parameters
<i>LABEL</i>	NTCLEAVE	<i>PARAM1 , PARAM2 , PARAM3 , PARAM4</i>

LABEL

This is the standard macro label for this request.

PARAM1 through PARAM4

These optional parameters may be included as needed by the calling module. When control is returned to the calling module, registers 1 through 4 respectively contain the values specified in param1 through param4. If a parameter value is omitted, the contents of the respective register are not modified. Parameter parameters may have the form (register) or address. The address must be RX instruction addressable.

The macro generates the following code:

label	NTCLEAVE ,	
	ICM R14,B'1111' ,	SAVEBACK GET ADDRESS OF PREV SAVE AREA
	BZ NL0011	SKIP TO FINISH NIT
	LR R2,R13	ADDRESS OF SAVE AREA TO FREE
	L R13,SAVEBACK	GET OLD SAVE AREA ADDRESS
	ST R13,NITSAVE	RESTORE NITSAVE TO OLD AREA
	L R0,ROOTSAFR	GET FREE CHAIN POINTER
	ST R0,0(R2)	CHAIN BLOCK INTO FREE CHAIN
	ST R2,ROOTSAFR	DITTO
	LM R14,R12,SAVER14	RESTORE ALL REGISTERS
	BR R14	RETURN TO THE CALLER
NL0011	L 15,=V(NXMDSPCH)	
	BR R15	AND GO
*		

Copy Members

Copy member NTCREGS provides register equates in the form of R1, R2 and must be copied into the user exits. In addition, members NTCROOT, NTCNIT, and NTCSAVE must also be copied into the exits. The NTCCXWA copy member can be copied into the disconnection exit if needed.

All of these copy members exist in the provided dummy exits.

Available Services

Since the exits are part of the NetEx load module, any NetEx service routine can be used.

NOTE: If any NetEx services are invoked from a user exit, the exit routine must be re-entrant. The following routines are described in this section:

- Getting Memory
- Freeing Memory
- Requesting Time of Day Services
- Requesting Timer Services

Getting Memory

If the NetEx provided 100 byte connection work area is not large enough, additional storage can be requested from NetEx. The form of these storage requests is shown below:

Label	Macro	Required Parameter	Optional Parameter
<i>LABEL</i>	NTCCALL	MEMGM	, <i>PARAM1</i>

LABEL

This is generated at the first executable address in the macro code.

MEMGM

Specifies the name of the NetEx memory manager.

PARAM1

Represents the size of the storage request, and is passed to the memory manager in register 1. It must be in (register) or RX-address format. The address of the storage obtained is returned in register 2.

The user is responsible for freeing all memory obtained in this manner.

Freeing Memory

If the user exit obtained additional NetEx memory, then it is responsible for freeing this storage. The form of these storage requests is shown below:

Label	Macro	Required Parameter	Optional Parameters
<i>LABEL</i>	NTCCALL	MEMFR	, <i>PARAM1</i> , <i>PARAM2</i>

LABEL

This is generated at the first executable address in the macro code.

MEMFR

Specifies the name of the NetEx memory manager.

PARAM1 and PARAM2

Represent the size and address, respectively, of the storage to free, and are passed to the memory manager in registers 1 and 2. They must be in (register) or RX-address format.

Requesting Time of Day Services

NetEx time of day services can be requested from a user exit by calling the NetEx timer manager. The form of these storage requests is shown below:

Label	Macro	Required Parameters	Optional Parameters
<i>LABEL</i>	NTCCALL	TMRTD	, <i>PARAM1</i> , <i>PARAM2</i>

LABEL

This is generated at the first executable address in the macro code.

TMRTD

Specifies the name of the NetEx timer manager.

PARAM1 and PARAM2

Identify the type of timer request. These parameters are described below:

PARAM2 = minus

Return the relative number of microseconds from the time NetEx began execution in the eight-byte area pointed to by *PARAM1*.

PARAM2 = 0

Return the current time of day in the format hh.mm.ss in the eight-byte area pointed to by PARAM1.

PARAM2 = 1

Return the current date in the format mm/dd/yy in the eight-byte area pointed to by PARAM1.

PARAM2 = 2

Return the current date in the format mm/dd/yy in the eight area pointed to by param1, and return the current time in the format hh.mm.ss in the eight byte area pointed to by param1+8.

Both PARAM1 and PARAM2 must be in (register) or RX-address format.

Requesting Timer Services

NetEx timer wait services can be requested from a user exit by calling the NetEx timer manager. The form of these storage requests is shown below:

Label	Macro	Required Parameter
<i>LABEL</i>	NTCCALL	TMRWT

LABEL

This is generated at the first executable address in the macro code.

TMRWT

Specifies the name of the NetEx timer manager.

The number of microseconds to wait should be stored in the eight-byte field located at NITORVAL in the current NIT.

Execution of the current NIT is suspended until the time specified has elapsed.

Miscellaneous Considerations

Most z/OS operating system services can be used from the user exits, with the exception of STIMER requests. A STIMER request from a user exit cancels the timer control that NetEx uses. If you want timing services, use the NetEx time management facilities previously described.

If the user exits collect data, the data can be written to a file. Accumulate or generate the data in the NetEx provided connection work area, and write the record to a file during one of the exits.

NOTE: The I/O process from the user exit may have an effect on NetEx performance. Remember to insert code in the exits to handle common types of error conditions encountered in the file.

Examples

The following examples indicate the type of processing that can be done with the user exits:

- Initialization/Termination Exit
- Offer/Connect Request Exit
- Connection Exit

These examples are contained on file 7 of the installation tape.

Initialization/Termination Exit

This exit initializes a user validation table and stores the address at ROOTUSER. This table is then available for reference from subsequent exits.

	TITLE 'MVS/VM NETEX REL 2.2 -- NXMUX1 -- USER EXIT MODULE'		
	COPY NTCOPT		

*			
*	MODULE NAME: NXMUX1		
*			
*	FUNCTION : PROVIDE USER EXIT SUPPORT		
*	DURING NETEX INITIALIZATION		
*	AND TERMINATION		
*			
*	ENTRY POINTS: NXMUX1IN - NETEX INITIALIZATION		
*	NXMUX1TM - NETEX TERMINATION		
*			
*	ENTRY CONDITIONS:		
*			
*	R8 - ROOT CONTROL BLOCK		
*	R9 - CURRENT NIT		
*	R13 - SAVE AREA ADDRESS		
*	R14 - RETURN ADDRESS		
*	R15 - ENTRY ADDRESS		
*			
*	EXIT CONDITIONS:		
*			
*	NONE		
*			

	SPACE 2		
UX1	NTCMOD MVS		
	EJECT		

*			
*	ENTRY: NXMUX1IN		
*			
*	FUNCTION:		
*			
*	THE PURPOSE OF THIS EXIT IS TO PERFORM USER		
*	INITIALIZATION PROCESSING.		
*			
*	IT WILL USE THE NETEX MEMORY MANAGER TO OBTAIN		
*	STORAGE IN WHICH TO PLACE A USER VALIDATION		
*	TABLE. THE ADDRESS OF THIS TABLE WILL BE STORED		
*	IN THE USER FIELD OF THE ROOT CONTROL BLOCK.		
*			
*	SUBSEQUENT CONNECTION EXITS CAN REFERENCE THIS		
*	TABLE TO VALIDATE NETEX REQUESTS.		
*			

	SPACE 1		
UX1IN	NTCENTER		
	LA R1,VTBLSIZE	SIZE OF STORAGE REQUIRED	
	NTCCALL MEMGM, (R1)	GET STORAGE	

```

SPACE 1
USING VTABLED,R2          ADDRESSABILITY TO TABLE
SPACE 1
MVC  VTABLED(VTBLSIZE),VTABLE  INITIALIZE IT
ST   R2,ROOTUSER          SAVE ADDRESS IN ROOT
NTCLEAVE ,                RETURN TO CALLER
EJECT

*****
*
*
*   ENTRY:NXMUX1TM
*
*   FUNCTION:
*
*   THE PURPOSE OF THIS EXIT IS TO PERFORM USER
*   TERMINATION PROCESSING.
*
*****

UX1TM  SPACE 1
       NTCENTER NOSAVE
       NTCLEAVE ,          RETURN TO CALLER
       EJECT

*****
*
*   CONSTANTS
*
*****

VTABLE SPACE 1
       DS      0H          CONNECTION VALIDATION TABLE
       DC      CL8'REILANDO',CL8'SERVER',CL8'MVS4341'
       DC      CL8'REILANDC',CL8'MVS4341',CL8'ACCOUNT1'
       DC      CL8'REILANDX',CL8'SERVERX',CL8'MVS4341'
       DC      CL8'REILANDC',CL8'MVS4341',CL8'ACCOUNT2'
       DC      CL8'REILANDO',CL8'SERVER',CL8'MVS4341'
       DC      CL8'REILANDB',CL8'MVS4341',CL8'ACCOUNT1'
       DC      CL8'REILAND1',CL8'SERVER1',CL8'MVS4341'
       DC      CL8'REILAND2',CL8'MVS4341',CL8'ACCOUNT1'
       DC      CL8'JOB2',CL8'OFFER2',CL8'HOST2'
       DC      CL8'JOBC',CL8'HOST1',CL8'ACCOUNT3'
       DC      X'FF'        END OF TABLE
VTBLSIZE EQU *-VTABLE      SIZE OF VALIDATION TABLE
EJECT

*****
*
*   DSECTS
*
*****

VTABLED SPACE 1
DSECT CONNECTION VALIDATION DSECT
VOFRJNM DS CL8 OFFERING JOB NAME
VOFRONM DS CL8 ,OFFER NAME
VOFRHNM DS CL8 ,AND HOST NAME
VCONJNM DS CL8 CONNECTING JOB NAME
VCONHNM DS CL8 AND HOST NAME
VACCTNO DS CL8 ACCOUNT NUMBER
VENTSIZE EQU *-VTABLED    SIZE OF EACH ENTRY
SPACE 1

```

VEOT	EQU	X'FF'	END OF VALIDATION TABLE INDICATOR
	SPACE	1	
	COPY	NTCREGS	REGISTER EQUATES
	COPY	NTCNIT	GENERATE NIT DSECT
	COPY	NTCSAVE	GENERATE SAVE AREA DSECT
	COPY	NTCROOT	GENERATE ROOT DSECT
	END		

Figure 47. Initialization/Termination Exit

Offer/Connect Request Exit

This exit uses the user validation table that was initialized by the Initialization exit. It checks for the validity of the requestor and passes the appropriate return code back to NetEx.

TITLE 'MVS/VM NETEX REL 2.2 -- NXMUX2 -- USER EXIT MODULE'				
COPY NTCOPT				

*				*
*	MODULE NAME: NXMUX2			*
*				*
*	FUNCTION :	PROVIDE USER EXIT SUPPORT		*
*		FOR VALIDATING NETEX		*
*		CONNECTION ATTEMPTS		*
*				*
*	ENTRY POINTS:NXMUX2OF - OFFER VALIDATION			*
*		NXMUX2CN - CONNECT VALIDATION		*
*				*
*	ENTRY CONDITIONS:			*
*				*
*		R1	- CWA PARM LIST ADDRESS	*
*		R2	- CURRENT NRB	*
*		R8	- ROOT CONTROL BLOCK	*
*		R9	- CURRENT NIT	*
*		R13	- SAVE AREA ADDRESS	*
*		R14	- RETURN ADDRESS	*
*		R15	- ENTRY ADDRESS	*
*				*

	SPACE 2			
UX2	NTCMOD MVS			
	SPACE 1			
	USING STAUXXWA,R1	PARM LIST ADDRESSABILITY		
	USING VTABLED,R7	VALIDATION TABLE ADDRESSABIL		
	USING NRB,R11	NRB ADDRESSABILITY		
*				
	EJECT			

*				*
*	ENTRY:NXMUX2OF			*
*				*
*	FUNCTION:			*
*				*
*	THE PURPOSE OF THIS EXIT IS TO PERFORM USER			*
*	VALIDATION OF NETEX OFFER REQUESTS.			*
*				*

```

*      IT WILL USE THE USER VALIDATION TABLE THAT      *
*      WAS INITIALIZED IN THE INITIALIZATION EXIT.      *
*
*      IF THE USER JOB NAME AND OFFER NAME EXISTS IN THE *
*      TABLE AND THE HOST NAME CORRESPONDS TO OUR HOST, *
*      THEN A ZERO RETURN CODE IS RETURNED FROM THE EXIT. *
*      OTHERWISE, A RETURN CODE OF 16 IS GENERATED, AND   *
*      A 9005 STATUS CODE IS RETURNED IN REGISTER 2.      *
*      NETEX WILL REFLECT THE 9005 IN THE USER'S NRBSTAT. *
*
*      ADDITIONAL REGISTER USAGE:
*
*          R7  - USER VALIDATION TABLE
*          R10 - CONNECTION WORK AREA
*          R11 - CURRENT NRB
*
*      EXIT CONDITIONS:
*
*          R1  = 0: ACCEPT REQUEST
*              = 12:REJECT REQUEST,
*                  RETURN 9000 IN NRBSTAT
*              = 16:REJECT REQUEST,
*                  RETURN R2 IN NRBSTAT
*
*****
*
EJECT
UX2OF  NTCENTER
      SPACE 1
      ICM  R7,15,ROOTUSER      GET VALIDATION TABLE
      BZ   OFFROK              WHERE IS IT ?
      L    R10,STAUCXWA        GET CONNECTION WORK AREA
      L    R11,STNRBADR        GET NRB ADDRESS
      LA   R11,00(,R11)        CLEAR EOL INDICATOR
OCHKTBL DS  0H
      CLI  VOFRJNM,VEOT        END OF TABLE ?
      BE   OFFRBAD             YES, REJECT REQUEST
      CLC  VOFRJNM,NITUSER     IS THIS ENTRY US ?
      BNE  ONXTENT             NO, GET NEXT ENTRY
      CLC  VOFRONM,NRBPNAME    IS THE OFFER NAME VALID ?
      BNE  ONXTENT             NO, GET NEXT ENTRY
      CLC  VOFRHNM,ROOTHOST    ARE WE ON CORRECT HOST ?
      BE   OFFROK             YES, ACCEPT IT
ONXTENT DS  0H
      LA   R7,VENTSIZE(,R7)    NEXT TABLE ENTRY
      B    OCHKTBL            CONTINUE LOOKUP

OFFRBAD DS  0H
      LA   R1,16               SET RETURN CODE
      L    R2,RC9005           AND NRBSTAT
      B    ODONE              ALL DONE
OFFROK  DS  0H
      SR   R1,R1              INDICATE GOOD RETURN CODE
ODONE   DS  0H
      NTCLEAVE (R1),(R2)      RETURN TO CALLER
EJECT
*****

```



```

*
* ENTRY:NXMUX2CN
*
* FUNCTION:
*
*     THE PURPOSE OF THIS EXIT IS TO PERFORM USER
*     VALIDATION OF NETEX CONNECT REQUESTS.
*
*     IT WILL USE THE USER VALIDATION TABLE THAT
*     WAS INITIALIZED IN THE INITIALIZATION EXIT.
*
*     IF THE USER JOB NAME EXISTS IN THE TABLE AS A VALID
*     CONNECTING JOB NAME FOR THIS HOST, AND THE DESTINATION
*     OFFER NAME AND HOST CORRESPONDS TO THE ENTRIES IN
*     THE TABLE, THEN A ZERO RETURN CODE IS RETURNED TO NETEX.
*     OTHERWISE, A 9015 STATUS CODE IS RETURNED IN REGISTER 2.
*     NETEX WILL REFLECT THE 9015 IN THE USER'S NRBSTAT.
*
* ADDITIONAL REGISTER USAGE:
*
*     R7  - USER VALIDATION TABLE
*     R10 - CONNECTION WORK AREA
*     R11 - CURRENT NRB
*
* EXIT CONDITIONS:
*
*     R1  = 0: ACCEPT REQUEST
*         = 12:REJECT REQUEST,
*             RETURN 9010 IN NRBSTAT
*         = 16:REJECT REQUEST,
*             RETURN R2 IN NRBSTAT
*
*****
*
UX2CN      EJECT
           NTCENTER
           SPACE 1
           ICM   R7,15,ROOTUSER      GET VALIDATION TABLE
           BZ    CONNOK              WHERE IS IT ?
           L     R10,STAUCXWA        GET CONNECTION WORK AREA
           L     R11,STNRBADR        GET NRB ADDRESS
           LA    R11,00(,R11)        CLEAR EOL INDICATOR

CCHKTBL   DS     0H
           CLI    VOFRJNM,VEOT      END OF TABLE ?
           BE     CONNBAD            YES, REJECT REQUEST
           CLC    VCONJNM,NITUSER   IS THIS ENTRY US ?
           BNE    CNXTENT            NO, GET NEXT ENTRY
           CLC    VCONHNM,ROOTHOST  ARE WE ON THE CORRECT HOST ?
           BNE    CNXTENT            NO, GET NEXT ENTRY
           CLC    VOFRONM,NRBPNAME  IS THIS A VALID CONNECTION ?
           BNE    CNXTENT            NO, GET NEXT ENTRY
           CLC    VOFRHNM,NRBHNAME  IS THIS A VALID CONNECTION ?
           BE     CONNOK            YES, ACCEPT IT

CNXTENT   DS     0H
           LA     R7,VENTSIZE(,R7)  NEXT TABLE ENTRY
           B      CCHKTBL           CONTINUE LOOKUP

```

```

CONNBAD  DS    0H
          LA    R1,16          SET RETURN CODE
          L     R2,RC9015      AND NRBSTAT
          B     CDONE          ALL DONE
CONNOK   DS    0H
          SR    R1,R1          INDICATE GOOD RETURN CODE
CDONE    DS    0H
          NTCLEAVE (R1),(R2)   RETURN TO CALLER
          EJECT
*****
*
*          CONSTANTS, DSECTS
*
*****
          SPACE 1
RC9005   DC    F'9005'
RC9015   DC    F'9015'
          LTORG
*
          SPACE 1
VTABLED  DSECT                CONNECTION VALIDATION DSECT
VOFRJNM  DS    CL8             OFFERING JOB NAME
VOFRONM  DS    CL8             ,OFFER NAME
VOFRHNM  DS    CL8             ,AND HOST NAME
VCONJNM  DS    CL8             CONNECTING JOB NAME
VCONHNM  DS    CL8             AND HOST NAME
VACCTNO  DS    CL8             ACCOUNT NUMBER
VENTSIZE EQU    *-VTABLED      SIZE OF EACH ENTRY
VEOT     EQU    X'FF'          END OF TABLE
          SPACE 1
          COPY  NTCREGS        REGISTER EQUATES
          COPY  NTCNIT         GENERATE NIT DSECT
          COPY  NTCSAVE        GENERATE SAVE AREA DSECT
          COPY  NTCROOT        GENERATE ROOT DSECT
          COPY  NTCCXWA        CONNECTION WORK AREA
          NRBD    ,            GENERATE NETEX REQUEST BLOCK
          END

```

Figure 48. Offer/Connect Request

Connection Exit

This exit uses the user validation table that was initialized by the Initialization exit. Following the completion of an Offer request, it checks for the validity of the connection and passes the appropriate return code back to NetEx. Figure 49 shows an example of a connection exit.

```

          TITLE 'MVS/VM NETEX REL 2.2 -- NXMUX3 -- USER EXIT MODULE'
          COPY  NTCOPT
*****
*
* MODULE NAME: NXMUX3
*
* FUNCTION :   PROVIDE USER EXIT SUPPORT
*              FOR VALIDATING NETEX
*              CONNECTIONS
*
* ENTRY POINTS: NXMUX3CX - CONNECTION VALIDATION

```

```

*
* ENTRY CONDITIONS:
*
*          R1  - CWA PARM LIST ADDRESS
*          R2  - CURRENT NRB
*          R3  - IF OFFER COMPLETED,
*                  ADDRESS OF 8 BYTE CONNECTING HOST NAME
*                  FOLLOWED BY 8 BYTE CONNECTING USER
*                  IF CONNECT COMPLETED,
*                  MINUS
*          R8  - ROOT CONTROL BLOCK
*          R9  - CURRENT NIT
*          R13 - SAVE AREA ADDRESS
*          R14 - RETURN ADDRESS
*          R15 - ENTRY ADDRESS
*
*****
      SPACE 2
UX3    NTCMOD MVS
*
      SPACE 1
      USING STAUCXWA,R1          PARM LIST ADDRESSABILITY
      USING VTABLED,R7          VALIDATION TABLE ADDRESSABIL
      USING NRB,R11             NRB ADDRESSABILITY
      EJECT
*****
*
* ENTRY:NXMUX3CX
* FUNCTION:
*   THE PURPOSE OF THIS EXIT IS TO PERFORM USER
*   VALIDATION OF NETEX CONNECTIONS.
*
*   IT WILL USE THE USER VALIDATION TABLE THAT
*   WAS INITIALIZED IN THE INITIALIZATION EXIT.
*
*   FOR CONNECTIONS RESULTING FROM THE COMPLETION OF
*   AN OFFER, THE ENTRY THAT CORRESPONDS TO OUR OFFER
*   REQUEST WILL BE FOUND IN THE USER VALIDATION TABLE.
*   IF THE CONNECTING USER AND HOST DO NOT EXIST AS
*   VALID CONNECTORS, THEN A RETURN CODE OF 8 WILL BE
*   RETURNED TO THE REQUEST DIRECTOR, AND A 9125 WILL
*   BE RETURNED IN REGISTER 3.  NETEX WILL ISSUE
*   A DISCONNECT TO THE REMOTE CONNECTOR, REFLECTING
*   THE 9125.  THE ORIGINAL OFFER REQUEST WILL BE
*   RE-ISSUED.
*
* ADDITIONAL REGISTER USAGE:
*          R7  - USER VALIDATION TABLE
*          R10 - CONNECTION WORK AREA
*          R11 - CURRENT NRB
*
* EXIT CONDITIONS:
*          R1  = 0: ACCEPT CONNECTION
*              = 4: REJECT CONNECTION,
*                  RETURN 9120 IN NRBSTAT,
*                  REISSUE OFFER
*              = 8: REJECT CONNECTION,
*                  RETURN R3 IN NRBSTAT,
*                  REISSUE OFFER

```

```

*           = 12:REJECT CONNECTION,
*           RETURN 9020 IN NRBSTAT,
*           DISCONNECT REMOTE WITH 9120
*           = 16:REJECT CONNECTION,
*           RETURN R2 IN NRBSTAT,
*           DISCONNECT REMOTE WITH R3
*
*****
*
EJECT
UX3CX  NTCENTER
      ICM  R7,15,ROOTUSER      GET VALIDATION TABLE
      BZ   CNXOK                WHERE IS IT ?
      L    R10,STAUCXWA        100-BYTE WORK AREA ADDR
      L    R11,STNRBADR        NRB ADDRESS
      L    R3,STPDAXX          SRB NAMES, X'FF'S, X'00'S
*
*   OFFER CONNECTION COMPLETION EXIT:
*   STPDAXX = SRB HOST, PROCESS NAMES ADDR
*   SREAD OF A CONFIRM:
*   STPDAXX = -1(ALL FF'S)
*   DRIVER CONNECT COMPLETE:
*   STPDAXX = 0
*
      SR    R1,R1              INITIALIZE RETURN CODE
      SR    R4,R4              0
      BCTR  R4,R0              -1(ALL FF'S)
      CR    R3,R4              SREAD CONFIRM?
      BE    DONE              YES ALL DONE
      LA    R3,00(,R3)        CLEAR FLAG BIT
      LTR   R3,R3              CHECK COMPLETION TYPE
      BZ    DONE              RETURN IF DRIVER(ALL 0'S)
CHKTBL DS    0H
      CLI  VOFRJNM,VEOT      END OF TABLE ?
      BE   CNXBAD            YES, REJECT CONNECTION
      CLC  VOFRJNM,NITUSER   IS THIS ENTRY US ?
      BNE  NXTENT            NO, GET NEXT ENTRY
      CLC  VOFRONM,NRBPNAME  IS IT OUR OFFER NAME ?
      BNE  NXTENT            NO, GET NEXT ENTRY
      CLC  VOFRHNM,ROOTHOST  IS IT OUR HOST NAME ?
      BE   CNXFOUND          YES, ACCEPT IT
NXTENT DS    0H
      LA   R7,VENTSIZE(,R7)  NEXT TABLE ENTRY
      B    CHKTBL            CONTINUE LOOKUP
CNXFOUND DS    0H
      CLC  VCONJNM,8(R3)     IS IT VALID USER CONNECTION ?
      BNE  NXTENT            NO, LOOK FOR ANOTHER ENTRY
      CLC  VCONHNM,0(R3)    IS IT VALID HOST CONNECTION ?
      BNE  NXTENT            NO, LOOK FOR ANOTHER ENTRY
CNXOK   DS    0H
      SR    R1,R1            INDICATE GOOD RETURN CODE
      B     DONE            ALL DONE
CNXBAD  DS    0H
      LA    R1,16            SET RETURN CODE
      L     R3,RC9150        AND NRBSTAT
      L     R2,RC9050        AND NRBSTAT
DONE    DS    0H

```

```

        NTCLEAVE (R1),(R2),(R3)    RETURN TO CALLER
        EJECT
*****
*
*          CONSTANTS, DSECTS
*
*****
        SPACE 1
RC9125   DC      F'9125'
RC9050   DC      F'9050'
RC9150   DC      F'9150'
        LTORG
*
        SPACE 1
VTABLED  DSECT          CONNECTION VALIDATION DSECT
VOFRJNM  DS      CL8     OFFERING JOB NAME
VOFRONM  DS      CL8     ,OFFER NAME
VOFRHNM  DS      CL8     ,AND HOST NAME
VCONJNM  DS      CL8     CONNECTING JOB NAME
VCONHNM  DS      CL8     AND HOST NAME
VACCTNO  DS      CL8     ACCOUNT NUMBER
VENTSIZE EQU      *-VTABLED  SIZE OF EACH ENTRY
VEOT     EQU      X'FF'      END OF TABLE
        SPACE 1
        COPY  NTCREGS      REGISTER EQUATES
        COPY  NTCNIT       GENERATE NIT DSECT
        COPY  NTCSAVE      GENERATE SAVE AREA DSECT
        COPY  NTCROOT      GENERATE ROOT DSECT
        COPY  NTCCXWA      CONECTION
        NRBD   ,           GENERATE NETEX REQUEST BLOCK
        END

```

Figure 49. Connection Exit

Appendix A. Glossary

This glossary contains terms that are used in this document. It includes frequently used acronyms and Network Executive Software product terminology.

ABEND

Abnormal end.

ACB

Access method control block.

ACK

Acknowledge.

adapter

A part that electronically or physically connects a device to a computer, another device or a communications line. A host interface connects to a computer, a link interface connects to a communications link.

address

A unique identifier assigned to a device connected to a network.

alternate path

Another NetEx network path an operation can use after a failure on a suspect path. See also alternate path retry (APR).

alternate path retry (APR)

A facility that allows a failed operation to be retried on another NetEx network path from the device performing the I/O operation. Alternate path retry improves the operation of NetEx remote hosts by switching network activity from a suspect or failing path to an alternate network path automatically. Preferred path routing in NetEx/IP detects the recovery of a previously failing primary NetEx network path and automatically restores network activity to that path.

ALTIO

Alternate I/O.

American Standard Code for Information Interchange (ASCII)

A standard that defines the codes for a character set to be used for information interchange between equipments of various manufactures and is the standard for digital communications.

APAR

Authorized program analysis report.

APF

Advanced function printing.

APR

Alternate path retry.

ASCII

American Standard Code for Information Interchange.

asynchronous

A class of data transmission service whereby all requests for service contend for a pool of dynamically allocated bandwidth and response time.

ATTN

Attention.

BFX

Bulk file transfer.

BFXJS

Bulk file transfer job submitter.

BFXTI

Bulk file transfer initiator.

BFXTR

Bulk file transfer responder.

BIST

Built in self test. A set of comprehensive self tests built into numerous StorageTek products. The purpose of BIST is to verify as many aspects of a chassis as possible in a self-contained environment.

buffer

A contiguous block of memory allocated for temporary storage of information in performing I/O operations. Data is saved in a predetermined format. Data may be written into or read from the buffers.

buffer memory

Temporary storage areas. Data going to or coming from the chassis is temporarily stored here until transmission on the network is completed.

Bulk File Transfer (BFX)

A Network Executive Software application software package. BFX utility allows users of NetEx Communications software to move large quantities of data between similar or dissimilar types of processors on a NetEx/IP networks. Any one version of BFX is fully compatible with any other. BFX consists of three separate programs: BFX Transfer Initiater (BFXTI), BFX Transfer Responder (BFXTR) and BFX Job Submitter (BFXJS). BFX is a trademark of Network Executive Software.

channel

The channel subsystem facilities associated with a single channel path.

CMDPRE

Command prefix.

command list

A list of commands and statements designed to perform a specific task.

command prefix

A character that identifies a NetEx subsystem. Use this character when issuing commands to a specific NetEx subsystem.

Configuration Manager

A utility that parses a text NCT file into a PAM file.

control unit (CU)

A hardware unit that controls the reading, writing, or displaying of data for one or more input/output units.

CPU

Central processing unit.

CRC	Cyclic redundancy check.
CSW	Channel status word.
CTC	Channel-to-channel.
CTCA	Channel-to-channel adapter.
CU	Control unit.
CUA	Common user access or channel unit address.
CUE	Control unit end.
cyclic redundancy check (CRC)	An error detection scheme.
DDB	Driver interface data block.
DEFBI	Default buffer input.
DEFBO	Default buffer output.
device adapter	Connects a network to a device channel.
DREF	Driver reference.
EBCDIC	Extended binary-coded decimal interchange code.
EOF	End of frame.
ESCON	Enterprise Systems Connection. The ESCON channel is an S/390-type channel.
ESTAE	Extended specify task abnormal exit.
ESTP	ESTAE parameter list.
FCB	Forms control block.
FCC	Federal Communications Commission.

frame

The unit of transmission on a Network Executive Software network. It consists of heading information and data. All data is sent on the network in this special envelope. A frame can be up to 4K bytes in length.

global network addressing (GNA)

A 32-bit network addressing scheme. It uses a 16-bit network address mated to the 16-bit unit address to uniquely identify units on a NetEx/IP network.

GNA

Global network addressing. A 32-bit network addressing scheme. It uses a 16-bit network address mated to the 16-bit unit address to uniquely identify units on a NetEx/IP network.

HCD

Hardware configuration definition.

header

Control information. It is transmitted at the beginning of a message, segment, datagram, packet, or block of data.

host

A data processing system that is connected to the network and with which devices on the network communicate. In the context of Internet Protocol (IP), a host is any addressable node on the network; an IP router has more than one host address.

host adapter

An interface connecting a host channel and a network.

IBM

International Business Machines.

ID

Identifier.

initial program load (IPL)

The process of starting (or restarting) the operating system.

Internet Protocol (IP)

A protocol suite operating within the Internet as defined by the *Requests For Comment* (RFC). This may also refer to the network layer (level 3) of the protocol stack (the layer concerned with routing datagrams from network to network).

I/O

Input/output.

IOCP

Input/output configuration program.

IPL

Initial program load.

IRB

Interrupt request block.

ISO

International standards organization.

JCL

Job control language.

JES, JES2, JES3

Job entry system.

JESCT

JES control table.

link

May refer to the communications facility used to interconnect local networks.

logical partition (LPAR)

A facility that allows you to divide the resources of a processor so that multiple copies of an operating system may exist on the same physical processor.

LPAR

Logical partition.

MCH

Machine check handler.

message proper

A basic addressing mechanism (nine to 64 bytes of data) used in all NetEx/IP messages. The first eight bytes are the message header and are common to all messages directed to the NetEx/IP network. These bytes indicate the source and destination addresses of both the physical adapters and the logical devices to which they are attached.

MIH

Missing interrupt handler.

MVS

Multiple virtual storage.

N/A

Not applicable.

NCT

Network configuration table.

NESiGate™

An IBM xSeries, or other manufacturer's server, containing Network Executive Software's custom software and hardware components. NESiGate enables a NetEx connection to an IP network from a host CPU, and may also run offload NetEx. NESiGates are no longer supported.

Network Configuration Table (NCT)

An internal data structure that is used by the NetEx Configuration Manager program to store all the information describing the network.

NETEX

NETwork EXecutive.

NETEX Request Block (NRB)

An array of parameters that pass information between calling programs and NetEx.

network

(1) A collection of interconnected computer systems, terminals and front-end processors. (2) Refers to the portion of a GNA address represented by the second byte (byte 1 when reading left to right). This portion of the GNA address has a hexadecimal value in the range from X'01' to X'FF'. The combination of domain, network, unit, and subaddress represent the GNA address of a particular processor on a network.

NETwork EXecutive (NetEx)

A Network Executive Software family of software designed to use IP networks that allows two or more application programs to communicate. NetEx is tailored to each supported operating system, but can communicate with any other supported NetEx, regardless of operating system.

NetEx is a registered trademark of Network Executive Software.

NIT

NetEx internal task.

NRB

NetEx request block.

NREF

Network reference.

NUB

NetEx user block.

octet form

Information considered as a sequence of eight-bit bytes.

Open Systems Interconnection (OSI)

A seven-layer protocol stack defining a model for communications among components (computers, devices, people, and et cetera) of a distributed network. OSI was defined by the ISO.

path

A route that can reach a specific host or group of devices.

PDS

Partitioned data set.

PFX

Print file transfer.

POST

Power on self test. One particular mode of BIST, POST is initiated a power-up, and runs on the chassis under test without operator intervention, special codes, or the connection of extraneous test equipment. POST tests all major logic boards in the chassis as quickly as possible, with a high degree of failure detection, and isolation to the board level.

port

(1) The point of entry/exit for data transmission to/from various chassis. (2) Ports may be logical or physical. The physical point of entry/exit for I/O operations to/from chassis is located on the I/O panel. (3) I/O operations through some chassis occur through separate logical data paths sharing a single physical port. Each of these data paths operates independently, as if it were a separate physical path. The point of entry/exit to these data paths is the logical port.

PRB

Program request block.

Print File Transfer (PFX)

A utility software package. The PFX allows users of NetEx communications software to transfer print files between similar or dissimilar types of processors on networks. PFX contains the facilities to select files and to make the file format conversions necessary for proper printing on the receiving host. PFX is a trademark of Network Executive Software.

protocol

A formal set of rules governing the format, timing, sequencing, and error control of exchanged messages on a network. A protocol may be oriented toward data transfer over an interface, between two logical units directly connected, or an end-to-end basis between two users over a large, complex network.

PSA

Prefixed storage area.

PSAREG

Prefixed storage area register save area.

PUT

Program update tape.

RAM

Random access memory. A memory device which may be written into or read from, with the time required to do so being independent of the data storage location. Information in a RAM is lost when the RAM loses power.

RU

Request unit check with bus out.

SID

Session identifier.

SMF

System management facility.

SMP/E

System modification program extended.

SNA

System network architecture.

SRB

Service request block.

SREF

Session reference.

SSCVT

Subsystem communications vector table.

SSVT

Subsystem vector table.

TCP/IP

An acronym for Transmission Control Protocol/Internet Protocol. These communication protocols provide the mechanism for inter-network communications, especially on the Internet. The protocols are hardware-independent. They are described and updated through *Requests For Comment* (RFC). IP corresponds to the OSI network layer 3, TCP to layers 4 and 5.

TREF

Transport reference.

TSCH

Test subchannel.

TUB	Transport user block.
UCB	Unit control block.
UCW	Unit control word.
UIM	Unit information module.
unit	The portion of a GNA address represented by the third byte (byte 2 when reading left to right). This portion of the GNA address has a hexadecimal value in the range from X'01' to X'FF'. The combination of domain, network, unit, and subaddress represent the GNA address of a particular processor on a network.
unit information module (UIM)	A software representation of a physical piece of hardware (for example, a control unit or a device) that describes the operating or connection rules for attaching that hardware to the processor for HCD.
WTO	Write to operator.

Index

ALIAS	77	BITMINT	78
Alternate Path Retry (APR)	31	CFOTM	79
asynchronous	190	CMDNO	79
BACKOFF	78	CMDPRE	80
basic I/O flow	6	CONFIG	81
BITMINT	78	CONTIME	82
CFOTM	79	DDMAXMLT	82
CMDNO	79	DEADTIME	83
CMDPRE	80	DEFBI	84
CONFIG	81	DEFBO	84
configuration manager	191	DESCCD	85
command statements	67	DFRT	85
DESELECT	69	DIRE	86
EXIT	69	DISE	86
MAKEPAM	70	DOWD	87
SELECT	71	DQHB	88
CONTIME	82	DQLB	89
DDMAXMLT	82	DRIVAVG	89
DEADTIME	83	DRIVLIM	90
DEFBI	84	DRIVMAX	91
DEFBO	84	ENDINI1	91
DESCCD	85	FUBS	92
DESELECT	69	GTRACE	92
device definition examples	18	HOST NAME	93
DFRT	85	IDLEMAX	93
DIRE	86	IDLETIME	94
DISE	86	INACTIVE	94
DOWD	87	IOCOUNT	96
DQHB	88	LARNAME	96
DQLB	89	LOGCMD	97
DRIVAVG	89	MAXBI	97
driver sublayer services	10	MAXBO	98
DRIVLIM	90	MAXDDBQ	98
DRIVMAX	91	MAXDREAD	98
ENDINI1	91	MAXDWRT	99
EXIT	69	MAXODATA	100
freeing memory	179	MAXSEG	101
FUBS	92	MBFI	102
getting memory	179	MBFO	102
glossary	190	MFCP	103
GTRACE	92	MIHTM	104
HCM2	4	MKBS	104
host	193	MSGVLV	105
HOST NAME	93	MSGML	106
IDLEMAX	93	MSGTM	107
IDLETIME	94	MSGXL	108
INACTIVE	94	NTXOPER	110
initialization statements	73	OSAIIF	111
ALIAS	77	OSAA	111
BACKOFF	78	OUTCLASS	112

PAGENIT.....	112	IOCOUNT.....	96
PAGESAVE.....	113	LARNAME.....	96
PAGEXTRA.....	114	LOGCMD.....	97
PAGExxx.....	115	MAKEPAM.....	70
PDQNAME.....	116	MAXBI.....	97
PRFP.....	117	MAXBO.....	98
RDDP.....	118	MAXDDBQ.....	98
RDIL.....	118	MAXDREAD.....	98
READTIME.....	119	MAXDWIT.....	99
ROUTCD.....	120	MAXODATA.....	100
RQPH.....	120	MAXSEG.....	101
RQPL.....	121	MBFI.....	102
RQPS.....	122	MBFO.....	102
RRCI.....	123	MFCP.....	103
SESAVG.....	123	MIHTM.....	104
SESLIM.....	124	MKBS.....	104
SESMAX.....	124	MSGLVL.....	105
SMFTYPE.....	125	MSGML.....	106
SMWTM.....	125, 126	MSGTM.....	107
SRBNAME.....	126	MSGXL.....	108
SRCI.....	127	NESiGate.....	194
SRPH.....	127	NetEx	
SRPL.....	128	characteristics.....	5
SRPS.....	128	connections.....	5
SRUI.....	129	driver sublayer services.....	10
SSINAME.....	129	external interface.....	5
STRP.....	130	installation.....	39
SUBSYS.....	130	installation prerequisites.....	14
SWTNAME.....	131	internal interaction.....	5
TRACE.....	132	network layer services.....	10
TRCNO.....	134	session layer services.....	9
TRCSIZE.....	135	transport layer services.....	9
TRMNAME.....	135	NETEX	
TWOPART.....	136	installation statements.....	73
UDPP.....	136	NetEx and the ISO model.....	7
UIFNAME.....	137	network configuration table (NCT)	
USERNAME.....	137	OSA considerations.....	65
WDOGINT.....	139	Network Configuration Table (NCT).....	194
XCONF.....	140	network layer services.....	10
XCONN.....	141	NTCCALL macro.....	175
XCXN.....	142	NTCENTER macro.....	176
XDATAI.....	143	NTCLEAVE macro.....	177
XDATAO.....	144	NTCMOD macro.....	174
XDISC.....	145	NTCOPT macro.....	173
XINIT.....	146	NTXOPER.....	110
XOFFR.....	147	Open Systems Interconnection (OSI).....	195
XTERM.....	148	operator commands	
installation prerequisites.....	14	SET MULTI.....	109
Alternate Path Retry (APR).....	31	OSA1IF.....	111
device definition examples.....	18	OSAA.....	111
device I/O gen requirements.....	18	OUTCLASS.....	112
NESiGate considerations.....	18	PAGENIT.....	112
release tape information.....	40	PAGESAVE.....	113
SMF considerations.....	15	PAGEXTRA.....	114
installing NetEx.....	39	PAGExxx.....	115
Internet Protocol (IP).....	193	path.....	195

PDQNAME	116
PRFP	117
RDDP	118
RDII	118
READTIME	119
requesting time of day services.....	179
requesting timer services	180
ROUTCD	120
RQPH	120
RQPL	121
RQPS	122
RRCI	123
SELECT	71
SESAVG	123
SESLIM	124
SESMAX	124
session layer services	9
SET MULTIH	109
SMFTYPE	125
SMWTM	125, 126
SRBNAME	126
SRCI	127
SRPH	127
SRPL	128
SRPS	128
SRUI	129
SSINAME	129
STRP	130
SUBSYS	130
SWTNAME	131
TRACE	132
transport layer services	9
TRCNO	134
TRCSIZE	135

TRMNAME	135
TWOPART	136
UDPP	136
UIFNAME	137
user exit	
copy members	178
user exits	149
examples	180
general	150
installation	173
miscellaneous considerations.....	180
NXMUX1IN	152
NXMUX1TM	153
NXMUX2CF	159
NXMUX2CN	156
NXMUX2OF	154
NXMUX3CX	161
NXMUX4DI	164
NXMUX4DO	167
NXMUX5DC	170
programming considerations	173
specific	151
USERNAME	137
WDOGIN	139
XCONF	140
XCONN	141
XCXN	142
XDATAI	143
XDATAO	144
XDISC	145
XINIT	146
XOFFR	147
XTERM	148

