Hxx4 Secure NetEx/IP™

Release 1.5.6

Software Reference Manual

Revision Record

Revision	Description
1.0	Manual released.
1.0-1	Minor clarification to Appendix C: Debugging User Applications
1.1	Minor updates to messages. See MTU.
1.2	Move API info to another document; missing NRB error codes; new security parameters.
1.2-1	Added optional step in zOS installation process
1.3	Add SSL PROTOCOL keyword to choose cipher level; custom code conversion tables;
1.4	Add support for additional features for name resolution and inbound connections.
1.5	Add support to declare address resolution for inbound separately from outbound connections.
1.5.2	Add support for console commands to SNXMAP
1.5.3	Add option to stop connection attempts after a 3501 or 3920 NRBSTAT
1.5.3-1	Clarify H214 specific requirements for use with Secure transfers (TCP/IP, SSL)
1.5.6	Add new message and clarify nrbstats

You may submit written comments to:

Network Executive Software, Inc.
Publications Department
6450 Wedgwood Road N Suite 103
Maple Grove, MN 55311
USA

Comments may also be submitted over the Internet by addressing e-mail to:

support@netex.com

or, by visiting our web site at:

www.netex.com

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

Preface

This manual describes the Secure NetEx/IP™ software for supported operating systems.

The currently supported operating systems and specific Secure NetEx/IP™ products are:

H214 for the IBM z/OS operating systems on IBM z Systems

H304 for the Unisys OS2200 on Dorado platforms

H804 for the Linux/Oracle Linux operating system on x86 platforms

H624 for the IBM AIX operating systems on IBM Power Systems

“Chapter 1: Introduction”, “Chapter 2: Secure NetEx/IP and the ISO Model”, “Chapter 3: Secure NetEx/IP Session Services, and Chapter 7: Operator Interface.

“Appendix A: NRB Error Codes” includes a list and description of the error messages and codes issued by Secure NetEx/IP.

The remaining appendices are installation and configuration instructions for each platform.

Readers are not expected to be familiar with Secure NetEx/IP before using this manual. However, an understanding of programming and using the host operating system is required.

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 that are not described in this publication and 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 Trademarks and Products

Network Executive Software	**NetEx, Secure NetEx/IP, BFX,**
The Open Group	**UNIX**
IBM Corporation	**IBM, AIX, z System, Power Systems, z/OS, GSKit**
Linus Torvalds	**Linux**
Unisys Corporation	**Dorado, OS2200**
Oracle Corporation	**Oracle**

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

Installation information contained in this document is intended for use by experienced System Programmers.

Document Conventions

The following notational conventions are used in this document.

Format	Description
`displayed information`	Information displayed on a CRT (or printed) is shown in **`this font`**.
`user entry`	*`This font`* 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.
value	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.

Glossary

asynchronous: A class of data transmission service whereby all requests for service contend for a pool of dynamically allocated ring bandwidth and response time.

ASCII: Acronym for American National Standard Code for Information Interchange.

BFX: Bulk File Transfer; Network Executive family of file transfer products.

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.

code conversion: An optional feature in Secure NetEx/IP that dynamically converts the user data from one character set to another (for example, ASCII, EBCDIC, et cetera).

GSKit: The Global Security Kit (GSKit) is a required component for the Secure Socket Layer (SSL) enablement for Secure NetEx/IP on IBM zOS.

header: A collection of control information 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.

hostname: A unique name of a host or server. The rules for a host name state that the name should be a text string consisting of the letters A through Z (upper or lower case), digits 0-9, minus sign (-), and the period (.). Note, the period is only allowed as the last character of the host name if it is the delimiter of the full domain name (FQDN). No spaces are permitted as part of a name. At least one character must be an alphabetic character and the last character must not be a minus sign or period. NetEx/IP hostnames are limited to 8 characters, while IP Hostnames are limited to 63 characters.

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 this protocol stack (the layer concerned with routing datagrams from network to network).

ISO: Acronym for International Standards Organization.

link: (1) A joining of any kind of IP networks. (2) The communications facility used to interconnect two trunks/busses on a network.

Secure NETwork Executive/IP (NetEx/IP): A family of software designed to enable two or more application programs on heterogeneous host systems to privately communicate. Secure NetEx/IP is tailored to each supported operating system but can communicate with any other supported Secure NetEx/IP, regardless of operating system.

NetEx is a registered trademark of Network Executive Software.

Open Systems Interconnection (OSI): A seven-layer protocol stack defining a model for communications among components (computers, devices, people, etcetera) of a distributed network. OSI was defined by the ISO.

Secure Socket Layer (SSL): A is a standard security technology for establishing an encrypted link between a server and a client

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.

Contents

Figures

Tables

Chapter 1: Introduction

Network Executive Software's Secure NetEx/IP™ allows two or more application programs (which may be on different host computers) to privately communicate with each other at multi-megabit speeds. The Secure NetEx/IP family of software consists of different versions of Secure NetEx/IP for use with different operating systems, such as the versions for use with the various UNIX, zOS or OS2200 operating system hosts. All of these versions provide a common high-level interface to simplify programming requirements. Secure NetEx/IP utility programs are also available, such as the Secure Bulk File Transfer (BFX™).

Secure NetEx/IP Characteristics

Secure NetEx/IP centralizes network considerations for IP networks, into a single piece of software. The following sections describe the characteristics of the Secure NetEx/IP software.

- External interface
- Internal interaction
- Secure NetEx/IP connections
- Design flow efficiency and flexibility
- Basic I/O flow

External Interface

The Secure NetEx/IP external interface for the application programmer is common for all versions of Secure NetEx/IP. Secure NetEx/IP provides the same requests for use in the programs that call NetEx/IP. *However, Secure NetEx/IP is not interoperable with NetEx/IP.* The calling programs may be written in C or other high-level languages. Secure NetEx/IP for zOS also has an assembler program interface. Secure NetEx/IP 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.

Internal Interaction

The internal operations of all supported versions of Secure NetEx/IP are consistent and allow the different versions to interact freely. Thus, any program using Secure NetEx/IP may communicate with any other program on the network that is also using Secure NetEx/IP. When a Secure NetEx application initiates a session (offer/connect), Secure NetEx/IP utilizes secure open system services of the TCP stack to establish the Secure NetEx/IP session and privately transfer data between Secure NetEx/IP nodes. The default port used in the Secure NetEx/IP network is TCP 3919. Configured ephemeral ports are also used for the data transfer.

Note: Secure NetEx/IP's listen port (TCP 3919) and the configured ephemeral ports must be allowed through firewalls between Secure NetEx/IP nodes.

To facilitate communication between hosts of different manufacture, Secure NetEx/IP supports code conversion. Secure NetEx/IP software can perform code conversion.

Secure NetEx/IP Connections

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

Secure NetEx/IP can establish multiple private connections at one time and can allow one program to have multiple private connections simultaneously.

Secure NetEx/IP also supports private 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 Secure NetEx/IP design enables many applications on the same processor to share the use of the network facility. Programs calling Secure NetEx/IP can be written without regard to the other programs calling Secure NetEx/IP.

Once Secure NetEx/IP accepts data from the caller, Secure NetEx/IP must deliver the data to its destination. The Secure NetEx/IP on each host handles code conversion, flow control, and error recovery.

Basic I/O Flow

Figure 1 shows the basic I/O flow between two programs using Secure NetEx/IP. The calling program communicates with Secure NetEx/IP through the Secure NetEx/IP user interface. Secure NetEx/IP then uses the available system services and network hardware to communicate with the calling program on the other processor.

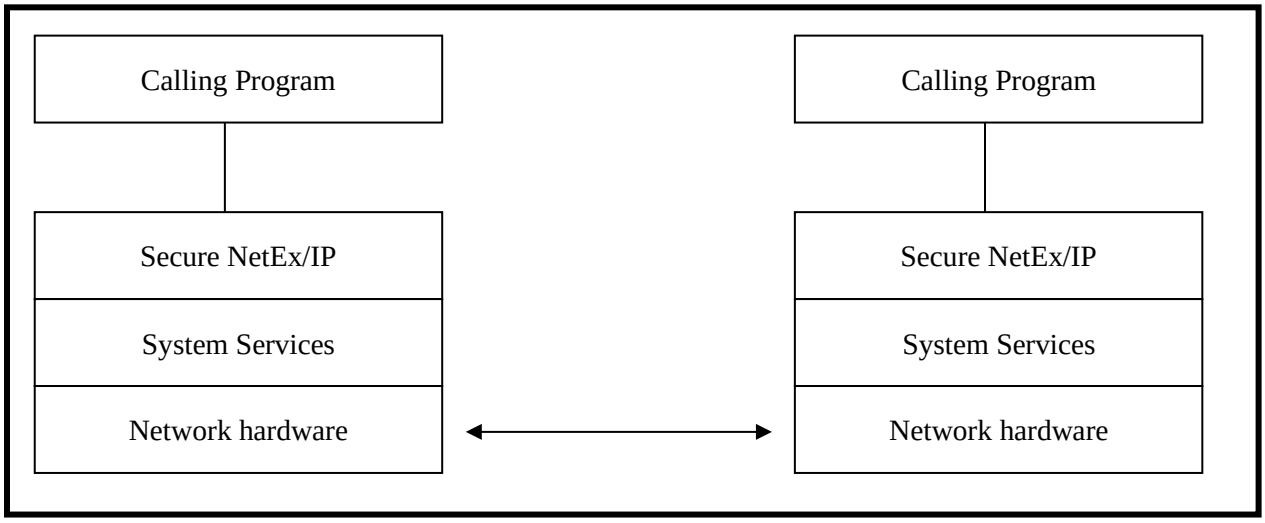

Figure 1. Basic I/O Flow

Secure NetEx/IP

Secure NetEx/IP consists of the user interface or API (linked with the user program) and SNXMAP, which is a separate program residing in each machine that Secure NetEx/IP users call on to perform common services for related to the Secure NetEx connections. Configuration of Secure NetEx/IP is done via the SNXMAP and its configuration file (snxmap.cfg). Each platform may have some unique configuration parameters. To understand them for your platform, refer to the appropriate installation section found in the Appendices and to Appendix G: Secure NetEx/IP Configuration File on page 95.

Chapter 2: Secure NetEx/IP and the ISO Model

In creating Secure NetEx/IP, Network Executive Software followed 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 Table 1). By using this modular structure, the internal function of each layer is self-contained and does not affect the functioning of other layers.

Table 1. ISO Model as Implemented

Layer	**Major Functions**
Application	High level description of data to be privately transferred and the destination involved
Presentation	Select data formats and syntax
Session	Establish a private session connection, report exceptions, and select routing
Transport	Manage data transfer and provide Secure NetEx/IP-to- Secure NetEx/IP 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.

Application	←----------→	Application
Presentation	←----------→	Presentation
Session	←----------→	Session
Transport	←----------→	Transport
Network	←----------→	Network
Data Link	←----------→	Data Link
Physical	←----------→	Physical
Physical Media (IP)		

Figure 2. ISO Model Communication

Note: The corresponding layers appear to communicate directly as indicated by the lines with arrows, but 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.

Session Layer Services

At the highest layer within Secure NetEx/IP (referring to the ISO model) the session layer software provides the general interface to the user's application/utility program. The Secure NetEx/IP session layer services include program-to-program connections using the best available network path, reading data, writing data, and disconnection. The user requests these services using a standard Secure NetEx/IP Request Block (NRB) (containing parameters), and the Secure NetEx/IP requests described in "Chapter 3: Secure NetEx/IP Session Services". The session layer software implements user requests by requesting services from the underlying layer.

Chapter 3: Secure NetEx/IP Session Services

The user interface to Secure NetEx/IP is a library that provides the user with access to the session functions of Secure NetEx/IP.

To communicate using the session layer of Secure NetEx/IP, the calling programs (that is, the programs that are calling Secure NetEx/IP) must utilize the user interface library to first establish a session connection. Once the session is established, private data transfer may take place in a variety of ways, depending on the needs of the calling programs. Once Secure NetEx/IP accepts data, the user is assured of its delivery (with the possible exception of catastrophic failures – for example, a machine going down or a major problem with the communication line). Sessions may be terminated by either of the parties at any time, although this should be done by mutual agreement.

The programming concepts and details are available in another document, by sending a request to support@netex.com.

Secure NetEx/IP Error Recovery Procedures

Calling programs must be able to recover from errors identified by Secure NetEx/IP. These errors will be returned when a Secure NetEx/IP operation does not complete successfully. The following paragraphs describe the Secure NetEx/IP error codes and some common error recovery procedures.

Error Codes

Whenever a Secure NetEx/IP request is issued, the results of the request are returned in one or both of two NRB fields, NRBSTAT and NRBIND. These are located at the beginning of the NRB to make their subsequent examination by high level language programs a simple matter.

NRBSTAT indicates whether an operation is in progress and whether it completed successfully or not. NRBIND indicates the type of information that arrived as the result of a read-type command (SREAD or SOFFER).

When an operation is accepted by the Secure NetEx/IP user interface, the value of NRBSTAT is set to the local value of –1. Thus, the sign of this word is an "operation in progress" flag for all implementations.

If an operation completed successfully, NRBSTAT will be returned as all zeroes. If a read-type command was issued, then an "indication" will be set in NRBIND when the SREAD completes.

If the operation did not complete successfully, the NRBSTAT will contain a standard error code. NRBSTAT is represented as a decimal number that is potentially as large as 2^{15} –1(32,767) The 2^{16} bit is not used so that it may remain an "in progress" flag for the 16-bit machines. The error codes are listed and described in "Appendix A: NRB Error Codes".

Common Error Recovery Procedures

The following items are some commonly encountered errors and an explanation of how to recover from them.

Other program not there – Operators or users must coordinate the running of the two Secure NetEx/IP programs so that one has not timed out before the other has had a chance to establish a session.

Other program busy – Retry Secure NetEx/IP after a suitable delay.

Secure NetEx/IP requests out of sequence – Sessions must be completely established before write or read requests can be issued. Sessions are established using the offer, connect, and confirm requests in that order.

Code Conversion

Secure NetEx/IP provides for common types of code conversion by using Secure NetEx/IP software facilities. The calling program uses the datamode (NRBDMODE) word of the NRB to specify code conversion. The caller simply specifies the source character set and the destination character set. Secure NetEx/IP then performs code conversion using software as necessary.

Secure NetEx/IP via TNP

Legacy NetEx Requester products can utilize a Secure NetEx/IP for encrypted connections across a WAN. This is accomplished by deploying Secure NetEx/IP in a LAN connected host which is also running TCP/NetEx Proxy (TNP). In this implementation, the requester products transport the NetEx requests and buffers between the NetEx applications and the Secure TNP. Secure TNP uses Secure NetEx/IP on behalf of the Requester connections to transfer the data securely to the remote Secure NetEx/IP and application.

Resolving NetEx/IP Requester Hosts to Use TNP

For TNP configurations, the NetEx/IP Requester hostname will need to resolve to use the Secure NetEx/TNP (i.e. Modify the DNS on the remote host to add the Netex Requester Hostname and TNP IP address). In the event the requester host name used by TNP is the same as the IP host name, NTXhostname can be used in DNS to properly route traffic to the TNP host. Secure NetEx will try to resolve NTXhostname before hostname when establishing a connection. At this time, the NetEx/IP Requester products are:

- H267IP for OpenVMS
- H367IP for HP NonStop
- H297IP for Bull Systems

Chapter 4: Operator Interface

There is a locally accessible operator interface for the SNXMAP component (snxmapop) which provides a small set of commands that allow you to display outstanding Secure NetEx offers and display current configuration parameters.

Command Descriptions Conventions

The following notational conventions are used in the Operator Command Help functions.

Format	Description
[ ]	One of the selections within the brackets are optional
…	One or more items may be specified

Command Line Mode

When you execute operator commands in command line mode, you initiate the SNXMAPOP program for each command and provide the command parameters as arguments to the program. To execute an operator command in command line mode, use the following general format:

```
snxmapop [option]… [command] [parameter]…
```

For the OS2200 system, modify the sample ECL in the release library to meet your dataset naming requirements (snxmapop/samp).

Operator Option Descriptions

This section details all the SNXMAPOP operator interface options. The following table briefly summarizes each option.

Operator Option	**Description**
-d	Enable debug output
-h, --help	Provide usage and a list of options and commands

Figure 3. SNXMAPOP Operator Options

Operator Command Descriptions

This section details all the SNXMAPOP operator interface commands. The following table briefly summarizes each command. ALL is the default if a command is not entered. Command minimal match is shown in upper-case.

Operator Command	Description
All	Show all registered offers (default)
DBGOFF	Disable debug messages in SNXMAP
DBGON	Enable debug messages in SNXMAP
Key	Show license key and status
Load	Load license key from key file
Parms	Show configuration parameters
SETConip	Set connect allow IP address
SETIp	Set host IP address
SHOWConip	Show connect allow IP addresses
SHOWIp	Show host IP addresses
UNSETConip	Unset connect allow IP address
UNSETIp	Unset host IP address

Figure 4. SNXMAPOP Operator Commands

ALL

Description

Displays a list of the pending SOFFERs in this SNXMAP waiting for SCONNECTs.

Format

snxmapop all

Examples

The following figure shows a sample ALL on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop all
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
snxmap offers=1

Name         Host       Port     Secure
--------     --------   -----    ---------
BFXJS        AIX3       37535    secure
```

Figure 5. ALL SNXMAPOP Command

PARMS

Description

Displays the current configuration parameters.

Format

snxmapop parms

Examples

The following figure shows a sample PARMS command output on an IBM AIX platform (output varies by platform and will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop parms
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
    BACKLOG=25
    CNVERIFY=OFF
    CONNTO=30
    DEBUG=OFF
    DEFBI=32768
    DEFBO=32768
    DNSRR=OFF
    FIPSMODE=ON
    IDLETO=6
    MAXBI=65535
    MAXBO=65535
    MULTIHOST=OFF
    RETRY-3501=ON
    RETRY-3920=ON
    SMWAIT=15
    LOG=0
    SYSLOG=1
    SYSLOGFAC=local6
    USECONIP=OFF
    USEDNS=ON
    USESETIP=OFF
    LCLHOST=AIX3
    CERTFILE=/usr/share/nesi/bfx/cert.pem
    KEYFILE=/usr/share/nesi/bfx/key.pem
    CAFILE=/usr/share/nesi/bfx/certs/trusted.pem
    CAPATH=/usr/share/nesi/bfx/certs
    SSLPROTO=tlsv1.2
    PORTNUM=1000
    PORTSTART=32000
```

Figure 6. PARMS SNXMAPOP Command

See “Appendix G: Secure NetEx/IP Configuration File” for defaults and meanings of the SNXMAP parameters.

KEY

Description

Displays the license key and status.

Format

snxmapop key

Examples

The following figure shows a sample KEY command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop key
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
License Key     : DGXI-YAA6-AAAW-U5UD-5YKX-SFLG
Expiration      : Fri Dec 01 23:59:59 CST 2017
Not Operational : Sun Dec 31 23:59:59 CST 2017
```

Figure 7. KEY SNXMAPOP Command

LOAD

Description

Load license key from key file and display the license key and status.

Format

snxmapop load

Examples

The following figure shows a sample LOAD command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop load
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
License Key     : DGXI-YAA6-AAAW-U5UD-5YKX-SFLG
Expiration      : Fri Dec 01 23:59:59 CST 2017
Not Operational : Sun Dec 31 23:59:59 CST 2017
```

Figure 8. LOAD SNXMAPOP Command

SETCONIP

Description

Set connect allow IP address.

Format

snxmapop setconip <ip-address>

Examples

The following figure shows sample SETCONIP command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop setconip 10.10.10.1
connecting to snxmap at 127.0.0.1:3919
snxmap status=0

aix3$ snxmapop setconip 10.10.10.256
connecting to snxmap at 127.0.0.1:3919
snxmap status=3988
Invalid IP address
```

Figure 9. SETCONIP SNXMAPOP Command

SETIP

Description

Set host IP address.

Format

snxmapop setip <host> <ip-address>

Examples

The following figure shows sample SETIP command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop setip ntxhost 10.10.10.1
connecting to snxmap at 127.0.0.1:3919
snxmap status=0

aix3$ snxmapop setip ntxhost10 10.10.10.2
connecting to snxmap at 127.0.0.1:3919
snxmapop: snxmap status=7
Hostname exceeds 8 characters

aix3$ snxmapop setip ntxhost 10.10.10.256
connecting to snxmap at 127.0.0.1:3919
snxmapop: snxmap status=3988
Invalid IP address
```

Figure 10. SETIP SNXMAPOP Command

SHOWCONIP

Description

Display a list of all IP addresses allowed to connect to this SNXMAP .

Format

snxmapop showconip

Examples

The following figure shows a sample SHOWCONIP command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop showconip
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
snxmap addresses=1

Connect IP Address
------------------
10.10.10.1
```

Figure 11. SHOWCONIP SNXMAPOP Command

SHOWIP

Description

Display a list of all hosts and IP addresses set in this SNXMAP .

Format

snxmapop showip

Examples

The following figure shows a sample SHOWIP command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop showip
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
snxmap hosts=3

Name      IP Address
--------  ---------------
HOST1     10.1.1.1
HOST1     10.1.1.3
NTXHOST   10.10.10.1
```

Figure 12. SHOWIP SNXMAPOP Command

UNSETCONIP

Description

Remove a connect allowed IP address.

Format

snxmapop unsetconip <ip-address>

Examples

The following figure shows a sample UNSETCONIP command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop unsetconip 10.10.10.1
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
```

Figure 13. UNSETCONIP SNXMAPOP Command

UNSETIP

Description

Remove a host IP address.

Format

snxmapop unsetip <host> <ip-address>

Examples

The following figure shows a sample UNSETIP command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop unsetip ntxhost 10.10.10.1
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
```

Figure 14. UNSETIP SNXMAPOP Command

DBGON

Description

Enable snxmap debug messages.

Format

snxmapop dbgon

Examples

The following figure shows a sample DBGON command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop dbgon
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
```

Figure 15. DBGON SNXMAPOP Command

DBGOFF

Description

Disable snxmap debug messages.

Format

snxmapop dbgoff

Examples

The following figure shows a sample DBGOFF command output on an IBM AIX platform. Output will vary based on configuration data (hostnames, IP addresses, keys...):

```
aix3$ snxmapop dbgoff
connecting to snxmap at 127.0.0.1:3919
snxmap status=0
```

Figure 16. DBGOFF SNXMAPOP Command

Appendix A: NRB Error Codes

When a NETEX request is issued, the results of the request are returned in one or both of two NRB fields, NRBSTAT and NRBIND. These fields are located at the beginning of the NRB to make their subsequent examination by high level language programs a simpler matter.

NRBSTAT is designed to indicate if an operation is in progress and whether it completed successfully. NRBIND is designed to indicate the type of information that arrived as the result of a read type command (OFFER or READ).

When the operation is accepted by the NETEX user interface, the value of NRBSTAT is set to –1. Thus, the sign of this word is an "operation in progress" flag for all implementations.

If an operation completed successfully, NRBSTAT is returned as all zeroes. If a read–type command was issued, then an "indication" is set in NRBIND. The termination of a session is always indicated by a disconnect indication in NRBIND regardless of the request type.

If the operation did not complete successfully, then NRBSTAT contains a standard error code. NRBSTAT is represented as four decimal digits. The thousands digit denotes the origin of the error; the low order three digits identify the error type. The codes for error origin are as follows.

Code	Description
0xxx	NETEX general. Errors detected by the user interface that prohibit proper process of the command.
1xxx	Reserved for Legacy NetEx Driver level errors (no longer issued)
2xxx	Transport level errors
3xxx	Session level errors.
4xxx	Reserved for Legacy NetEx Network level errors (no longer issued)
5xxx-8xxx	Reserved for future NETEX functions
90xx	Reserved for errors returned by user exits on the local host
91xx	Reserved for errors returned by user exits on the remote host during the connection process.
9200-32767	Reserved for future NETEX functions.

Table 2. Origin of NRB Error Codes

Note the following when using these codes:

Some errors cause the loss of the connection or result in a connection not being established. Any status code that implies that the connection is no longer useful has a 6 (Disconnect Indication) returned in NRBSTAT. Any attempts to issue further requests to that connection have a x100 (no Nref) error returned to it.

All errors that result in the loss of the connection and a Disconnect Indication in NRBIND are indicated by an asterisk (*) following the error code number.

Note: A 0000 in field NRBSTAT means successful completion of the NETEX request. A -1 means that the request is still in progress.

The following sections describe the errors in numerical order starting with general NETEX error, followed by driver, transport, and session level errors.

General Errors

The following errors are general NETEX errors.

0000	Successful completion
0001	A read type operation completed normally within NETEX, but the Pdata buffer provided by the user was not large enough to hold the data. NRBLEN and NRBUBIT reflect the amount of data the other party intended to send. However, the amount of data moved to the user's program was only the amount of addressable units specified in NRBBUFL. NRBIND specifies the type of data sent to the user. Requests affected: OFFER, READ. The status of the connection is not affected.
0004	The request code (NRBREQ) is not valid. The operation is ignored, and the status of the connection specified by NRBNREF is not affected
0005	The buffer size specified (in NRBBUFL for read and NRBLEN for write) exceeds an implementation defined NETEX maximum. The operation is suppressed. The status of the connection is not affected.
0006	Offer name was not specified or contains non-alphanumeric characters
0007	Host name was not specified or contains non-alphanumeric characters
0011	A read-type operation completed normally within NETEX, but the Odata buffer provided by the user was not large enough to hold the data. NRBPROTL reflects the amount of data the other party intended to send; however, the amount of data moved to the user's program was only the amount of addressable units originally specified in NRBPROTL. NRBIND specifies the type of data sent to the user. Requests affected: OFFER, READ. The status of the connection is not affected
0021	A read-type operation completed normally within NETEX, but BOTH the Odata and Pdata buffers were too small to hold the incoming data. NRBLEN/NRBUBIT and NRBPROTL reflect the amount of data the other party intended to send; however, the amount of data moved to the user's program was only the amount of addressable units originally specified in NRBLEN and NRBPROTL. NRBIND specifies the type of data sent to the user. Request affected: OFFER, READ. The status of the connection is not affected.
0310	The user has attempted to re-use an NRB before a previous request issued with that NRB has completed. The request will be rejected. When the original request issued with that NRB completes, then the NRB will be once more updated with the status of that request.

Table 3. General NRB Error Codes

License Specific Errors

0610	Can't read license
0611	License is invalid
0612	License has expired
0613	License does not support TNP

Table 4. License Specific NRB Error Codes

Transport Service Errors

2300	An SREAD request timed-out before a response was received on the network. The status of the connection is not affected.
2307	A SREAD request has been issued to a connection that is in the process of servicing a remote caller or NETEX initiated Disconnect. A Disconnect Indication is pending from NETEX.

Session Service Errors

3005	During a WRITE operation, the length of the buffer as specified by NRBLEN exceeds the maximum buffer size found in NRBBLKO. The WRITE operation is rejected. The connection remains outstanding.
3006	The length of PDATA sent on a CONNECT, CONFIRM, or DISCONNECT is greater than the maximum allowed. The request is rejected.
3100*	The Sref specified by NRBNREF is not in use or is not owned by this applications program. The request is rejected. The status of other connections owned by this application remains unchanged.
3101	On an SWRITE request, a DATAMODE was specified that is not supported on this host.
3201	A general TCP socket error occurred; review errors logged prior to this error for platform specific socket errors.
3202	Connection closed unexpectedly. Refer to the remote side for reasons why the connection was closed.
3300	An SOFFER request timed-out before a response was received on the network. When the SOFFER timed out, then the connection will not have taken place.
3390	NETEX received a TCP close with a 0 data length indicator
3402*	The remote application has failed to issue an SREAD request for a period of elapsed time (READTO) specified by the installation systems programmer on the remote host. The connection is terminated. A Disconnect Indication will be found in NRBIND.
3422	Session was disconnected by application. SOFFER terminates with a Disconnect Indication in NRBIND.
3500 *	A connect was sent to the remote host but no response was received for a period of elapsed time (CONNTO). The SCONNECT terminates with a Disconnect Indication in NRBIND.
3501*	The PNAME specified (with the specified security mode) is not OFFERed on the HOST specified during the SCONNECT. The SCONNECT terminates with a Disconnect Indication in NRBIND.
3503*	The number of user session connections permitted by NETEX has been exceeded. Session service cannot be offered at this time. The SCONNECT or SOFFER is rejected.
3506*	The HOST specified in an SCONNECT request does not exist anywhere on the network generated by the installation systems programmer. The SDISCONNECT terminates with a Disconnect Indication in NRBIND.
3509*	The specified value of NRBBLKO exceeds an installation or implementation defined maximum. The connection request is rejected.

3510*	The specified value of NRBBLKI exceeds an installation or implementation defined maximum. The connection request is rejected.
3910	A memory request for a code conversion table failed. No session can be established. NRBIND set to Disconnect indication.
3911	An error occurred while initializing code conversion tables. Possible problems are missing file, unable to access file, incomplete or bad file data. No session can be established. NRBIND set to Disconnect indication.
3920	On an SOFFER or SCONNECT the remote host did not allow a connect from the source IP address. No session can be established. NRBIND set to Disconnect indication
3969	A memory request for a request block failed. Internal NRB could not be allocated. Session is unchanged.
3970	A memory request for a work area failed. Failed to initialize Secure NetEx/IP API global work area. No session can be established. NRBIND set to Disconnect indication.
3971	A memory request for a thread mutex failed. Failed to initialize Secure NetEx/IP API global work area. No session can be established. NRBIND set to Disconnect indication.
3972	An initialization request or a thread mutex failed. Failed to initialize Secure NetEx/IP API global work area. No session can be established. NRBIND set to Disconnect indication.
3973	A memory request for pthread condition failed. Failed to initialize Secure NetEx/IP API global work area. No session can be established. NRBIND set to Disconnect indication.
3974	An initialization request for a thread condition failed. Failed to initialize Secure NetEx/IP API global work area. No session can be established. NRBIND set to Disconnect indication.
3975	On an SOFFER or SCONNECT, the session control block could not be allocated. No session can be established. NRBIND set to Disconnect indication.
3980	A request to SNXMAP failed to complete. The session is terminated with a NRBIND set to Disconnect indication.
3981	A request to SNXMAP failed with a return code. The session is terminated with a NRBIND set to Disconnect indication.
3982	A request to create a new thread failed with the return code. The session is terminated with a NRBIND set to Disconnect indication.
3987	SNXMAP failed a memory request. The session is terminated with a NRBIND set to Disconnect indication.
3988	SNXMAP received an invalid request. The session is terminated with a NRBIND set to Disconnect indication.
3989	The length in the protocol header was incorrect. Secure NetEx/IP protocol failed. For SOFFER, the session is terminated with a NRBIND set to Disconnect indication. Otherwise the session is unchanged.
3990	The version in the protocol header was incorrect. Secure NetEx/IP protocol failed. For SOFFER, the session is terminated with a NRBIND set to Disconnect indication. Otherwise the session is unchanged.

3991	The header type in the protocol header was incorrect. Secure NetEx/IP protocol failed. For SOFFER, the session is terminated with a NRBIND set to Disconnect indication. Otherwise the session is unchanged.
3992	An error occurred with creating the encrypted connection. SSL failure. Secure NetEx/IP protocol failed. For SOFFER, the session is terminated with a NRBIND set to Disconnect indication. Otherwise the session is unchanged.
3995	A memory request failed for the GSK parameters. SSL failure. Secure NetEx/IP protocol failed. For SOFFER, the session is terminated with a NRBIND set to Disconnect indication. Otherwise the session is unchanged.
3997	An error occurred relating to the TCPCFG file.

Table 5. Session Service NRB Error Codes

Appendix B: Secure NetEx Messages

This section contains a description of the messages issued by Secure NetEx/IP. These messages are displayed in the stdout/stderr file of the user's application if logging is set to the appropriate level. Message numbers are post fixed with a D, T, I or E to reflect the category as well. Refer to the specific platform installation/configuration Appendix for information on setting up messages.

Message numbers are categorized by the following:

SNX1x00D – Debug (not documented in this manual; for NetEx Support only)

SNX2x00T – Trace (not documented in this manual; for NetEx Support only)

SNX3xxxI – Informational

SNX4xxxE – Error

SNXxxxxC – Continuation of previous message

Messages may be prefixed with a date and timestamp of the following format if utilizing a logging facility:

Sun Jan 27 01:03:52 2017

Informational Messages

SNX3100I %s

Description: %s contains the version and tab level of the Secure NetEx/IP code

User Response: None

SNX3101I csokcntx(%d): Setting nsswitch failed

Description: On an IBM zOS system, an attempt to define the name resolver search did not complete successfully. The system search order will be used instead.

%d uniquely identifies the internal session ID in use

User Response: Ensure the host names being used are resolved as intended

SNX3102I csokcntx(%d): RFCONNECT doportmap GET to %s timed out

Description: While processing a CONNECT request, a response was not received from the remote SNXMAP facility. The next IP address returned from the name resolver will be tried.

%d uniquely identifies the internal session ID in use

%s identifies the IP address that did not respond

User Response: None

SNX3103I csokcntx(%d): RFCONNECT doportmap GET to %s error, errno=%d1

Description: On the connecting side, this message informs the user there was an error trying to connect to the remote SNXMAP facility to determine the OFFERing application port number. Previous error messages have more details.

%d uniquely identifies the internal session ID in use

%s IP address the remote NetEx hostname resolved to

%d1 System error when (trying to or) communicating with the remote host SNXMAP facility

User Response: Review previous error messages.

SNX3104I csokcntx(%d): RFCONNECT doportmap GET to %s error, status=%d1

Description: On a connect request initiating communication to the SNXMAP facility on a remote host returned an error status.

%d uniquely identifies the internal session ID in use

%s IP address the remote Netex hostname resolved to

%d1 the nrb status returned

User Response: Refer to NRB error codes for details regarding this issue.

SNX3105I csokcntx(%d): RFCONNECT securexfr=%d1, port=%d2

Description: Preparing to create the user connection.

%d uniquely identifies the internal session ID in use

%d1 transfer mode (0 – unsecure, 1 – secure)

%d2 remote offer port

User Response: None

SNX3106I csokcntx(%d): COMAPI=%i, addr=%x

Description: On a Unisys OS2200 system, displays the TCP configuration information for this session.

%d uniquely identifies the internal session ID in use

%i identifies the COMAPI being used

%x identifies the local IP address (i.e. 0x7F000001) where local SNXMAP facility is configured for

User Response: None

SNX3107I csokcntx(%d): Running RealTime Level = %i

Description: On a Unisys OS2200 system, displays the realtime level of the program if running in realtime mode.

%i identifies the realtime level

User Response: RTLVL is set in the TCPCFG configuration file, raise or lower as needed. 5 is the highest possible, 35 is the lowest level.

SNX3110I csokcntx(%d): RFCONNECT doportmap GETIP timed out

Description: While processing a CONNECT request, a response was not received from the local SNXMAP facility. The next IP address resolution method will be tried.

%d uniquely identifies the internal session ID in use

User Response: None

SNX3111I csokcntx(%d): RFCONNECT doportmap GETIP error, errno=%d1

Description: On the connecting side, this message informs the user there was an error trying to connect to the local SNXMAP facility to determine the OFFERing application IP address. Previous error messages have more details.

%d uniquely identifies the internal session ID in use

%d1 System error when (trying to or) communicating with the local host SNXMAP facility

User Response: Review previous error messages.

SNX3112I csokcntx(%d): RFCONNECT doportmap GETIP error, status=%d1

Description: On a connect request initiating communication to the SNXMAP facility on the local host returned an error status.

%d uniquely identifies the internal session ID in use

%d1 the nrb status returned

User Response: Refer to NRB error codes for details regarding this issue.

SNX3113I csokcntx(%d): environment variable 'SNETEX_SERVICE' set port %d1

Description: On a Unix system this shows that an SNXMAP port number was specified in the environment.

%d uniquely identifies the internal session ID in use

%d1 port that will be used to communicate with SNXMAP

User Response: None

SNX3114I csokcntx(%d): getservbyname '%s' returned port %d1

Description: Shows successful SNXMAP service name resolution.

%d uniquely identifies the internal session ID in use

%s service name

%d1 port that will be used to communicate with SNXMAP

User Response: None

SNX3115I csokcntx(%d): getservbyname '%s' returned NULL, using default port %d1

Description: Shows unsuccessful SNXMAP service name resolution. Default port will be used.

%d uniquely identifies the internal session ID in use

%s service name

%d1 port that will be used to communicate with SNXMAP

User Response: None

SNX3201I thread(%d): RFCONNECT rmtaddr=%s:%d1

Description: Connect request in progress; preparing for connect.

%d uniquely identifies the internal session ID in use

%s the remote IP address (the OFFER application)

%d1 the remote port number (the OFFER application)

User Response: None

SNX3202I thread(%d): setting secure opts

Description: On a Unisys OS2200 system, setup of secure options prior to establishing the connection to the remote host.

%d uniquely identifies the internal session ID in use

User Response: None

SNX3203I thread(%d): Non secure transfer

Description: On a Unisys OS2200 system, indicates using a non-secure connection (connect/client).

%d uniquely identifies the internal session ID in use

User Response: None

SNX3204I thread(%d): RFCONNECT lcladdr=%s:%d1

Description: Connect request in progress; socket is connected.

%d uniquely identifies the internal session ID in use

%s the remote IP address (the OFFER application)

%d1 the remote port number (the OFFER application)

User Response: None

SNX3205I thread(%d): RFCONNECT SSL connection using %s %s1

Description: On a RFCONNECT, on a Unix system, the secure socket is connected using the protocol and cipher description (connect/client).

%d uniquely identifies the internal session ID in use

%s the protocol description

%s1 the cipher description

User Response: None

SNX3206I thread(%d): RFOFFER request (%d1 secs)

Description: RFOFFER request processing.

%d uniquely identifies the internal session ID in use

%d1 Offer timeout set by the application

User Response: None

SNX3207I thread(%d): RFOFFER listening: addr=%s:%d1

Description: Offer request; prepare to do the listen.

%d uniquely identifies the internal session ID in use

%s the IP address (the OFFER application)

%d1 the port number (the OFFER application)

User Response: None

SNX3208I thread(%d): setting secure opts

Description: On a Unisys OS2200 system, setup of secure options prior to the listen (Offer/server).

%d uniquely identifies the internal session ID in use

User Response: None

SNX3209I thread(%d): Non secure transfer

Description: On a Unisys OS2200 system, not a secure connection (Offer/server).

%d uniquely identifies the internal session ID in use

User Response: None

SNX3210I thread(%d): RFOFFER accepted lcladdr=%s:%d1, rmtaddr=%s1:%d2

Description: Processing Offer when Connect is received.

%d uniquely identifies the internal session ID in use

%s the local IP address (the OFFER application)

%d1 the local port number (the OFFER application)

%s the remote IP address (the CONNECT application)

%d2 the remote port number (the CONNECT application)

User Response: None

SNX3211I thread(%d): RFOFFER SSL connection using %s %s1

Description: On a RFOFFER, on a Unix system, the secure socket is connected using the protocol and cipher description (offer/server).

%d uniquely identifies the internal session ID in use

%s the protocol description

%s1 the cipher description

User Response: None

SNX3212I thread(%d): realtime =%i rc=%x

Description: On a Unisys OS2200 system, the thread task is running realtime level.

%d uniquely identifies the internal session ID in use

%i identifies the realtime level

%x displays the return code from the set realtime request.

User Response: None

SNX3213I thread(%d): RFOFFER terminated lcladdr=%s:%d1, rmtaddr=%s1:%d2 stat=%d3

Description: Processing Offer when Connect is received.

%d uniquely identifies the internal session ID in use

%s the local IP address (the OFFER application)

%d1 the local port number (the OFFER application)

%s the remote IP address (the CONNECT application)

%d2 the remote port number (the CONNECT application)

%d3 the nrb status returned

User Response: None

SNX3301I Certificate is self-signed

Description: Self-signed certificates must be added to trust store (local verify).

User Response: None

SNX3302I Certificate subject name: %s

Description: Local certificate subject name field (local cert verify).

%s local certificate subject name

User Response: None

SNX3303I Peer certificate:

Description: This message indicates the beginning of peer certificate information. Additional messages will follow for items such as serial number, issuer, subject, expiration dates and X509V3 extensions.

User Response: None

SNX3309I No peer certificate

Description: No peer certificate received during SSL handshake

User Response: None

SNX3310I SSL check host: %s : %s1

Description: Performing peer certificate check for hostname or IP.

%s	hostname to check
%s1	IP address to check

User Response: None

SNX3311I SSL FIPS mode is %s

Description: SSL FIPS mode setting.

%s	enabled/disabled

User Response: None

SNX3401I show_cert(%d): id=%d1(%s)

Description: On IBM zOS system this is the certificate to be used for this connection.

%d	uniquely identifies the internal session ID in use
%d1	identifies the attribute in the certificate
%s	contents of the attribute in the certificate

User Response: None

SNX3402I show_cert(%d): elem_count=%d1

Description: On IBM zOS system this displays the number of the elements in the certificate to be used for this connection.

%d	uniquely identifies the internal session ID in use
%d1	number of elements in the certificate

User Response: None

SNX3403I show_cert(%d): i=%d1, cert_data_id=%d2, cert_data_l=%d3

Description: On IBM zOS system this displays the element id and data in the certificate to be used for this connection.

%d uniquely identifies the internal session ID in use

%d1 index of certificate element

%d2 the element identifier in the certificate

%d3 identifies the length of the element in the certificate

User Response: None

SNX3404I show_cert(%d): i=%d1, cert_data_id=%d2, cert_data_p='%s', cert_data_l=%d3

Description: On IBM zOS system this displays the attribute and data in the certificate to be used for this connection

%d uniquely identifies the internal session ID in use

%d1 index of certificate element

%d2 the element identifier in the certificate

%s the contents of the element in the certificate

%d3 identifies the length of the element in the certificate

User Response: None

SNX3405I show_cert(%d): no cert data

Description: On an IBM zOS system, there was no certificate for this connection.

%d uniquely identifies the internal session ID in use

User Response: You may need to verify with your site security that the NetEx Hostname (IP address) has security credentials.

SNX3406I show_cert(%d): gsk_attribute_get_cert_info failure, rc=%d1

Description: On an IBM zOS system, an error was returned when attempting to get the certificate info.

%d uniquely identifies the internal session ID in use

%d1 error returned on the call

User Response: Contact support@netex.com for details on the error. You may need to verify with your site security that the NetEx/IP Hostname (IP address) has security credentials.

SNX3407I find_cert_data(%d): slen(%d1) > dlen(%d2), using dlen

Description: On an IBM zOS system, when evaluating the certificate, the length of the element is longer than the size of the certificate.

%d uniquely identifies the internal session ID in use

%d1 length of the certificate element data

%d2 length of the supplied area for the element data

User Response: None

SNX3408I find_cert_data(%d): elem %d not found

Description: On an IBM zOS system, when evaluating the certificate, the element to verify is not found in the certificate.

%d uniquely identifies the internal session ID in use

%d1 the element to verify

User Response: You may need to verify with your site security that the NetEx Hostname (IP address) has security credentials.

SNX3409I find find_cert_data(%d): no cert data

Description: On an IBM zOS system, no certificate data was returned for this NetEx Hostname (IP Address).

%d uniquely identifies the internal session ID in use

User Response: You may need to verify with your site security that the NetEx Hostname (IP address) has security credentials.

SNX3501I doportmap(%d): read response select timed out

Description: SNXMAP select read timeout on the socket (no response from SNXMAP for 10 seconds)

%d uniquely identifies the internal session ID in use

User Response: None

SNX3509I SO_EXTERNLOOP errno=%i, optValue=%i (code location)

Description: Requesting an internal loopback, the Unisys system returned an error. The errno is displayed. Internal loopbacks will be disable and external loopbacks will be user. This requires CP18 level from Unisys.

%i is the errno that was returned.

%i is the value of the EXTERNALLOOP requested.

User Response: Upgrade the COMAPI to the CP18 level.

SNX3510I doportmap(%d): connect timeout

Description: A connect to a remote SNXMAP timed out after CONNTO seconds.

%d uniquely identifies the internal session ID in use

User Response: Verify the remote NetEx hostname is correct or the network path to the host is available.

SNX3900I TCPCFG warnings

Description: While processing the TCPCFG file on Unisys warnings were encountered. The warnings will be logged following this message.

User Response: None

SNX3902I SNXMAP IS RESPONDING TO KEYIN: %s\n

Description: Shows the console KEYIN snxmap is responding to on Unisys.

%s Keyin val

User Response: None

SNX3903I SNXMAP TERMINATION REQUESTED

Description: The SNXMAP program received a termination request from the Unisys Console.

User Response: This command may take up to 10 seconds to complete.

SNX3904I SNXMAP INVALID COMMAND

Description: The Unisys operator entered an unrecognized command.

User Response: Currently TERM and SWITCH are the only valid commands

SNX3908I USE= NOT CONFIGURED; SWITCH NOT AVAILABLE

Description: The USE = Parameter is either missing or not properly pointing to the print file in the ECL. The switch command will not function.

User Response: Add or correct the USE = statement to properly point to the use statement for the print file

SNX3909I USE INVALID RC=%i, RC2=%i

Description: While processing the USE= parameter, UCSINFO returned these status codes.

User Response: The USE = parameter is not correctly pointing to the break-pointed print file. Correct the USE= statement.

SNX3904I SNXMAP INVALID COMMAND

Description: The Unisys operator entered an unrecognized command.

User Response: Currently TERM is the only valid command

SNX3904I SNXMAP INVALID COMMAND

Description: The Unisys operator entered an unrecognized command.

User Response: Currently TERM is the only valid command.

Error Messages

SNX4101E csokcntx(%d): RFCONNECT doportmap LICCHK timed out

Description: While processing a CONNECT request, a response was not received from the local SNXMAP facility within 5s.

%d uniquely identifies the internal session ID in use

User Response: Verify the local SNXMAP is listening on the 'snetex' service port.

SNX4102E csokcntx(%d): RFCONNECT doportmap LICCHK socket error, errno=%d1: %s

Description: While processing a CONNECT request, a socket error was not received connecting to the local SNXMAP facility.

%d uniquely identifies the internal session ID in use

%d1 system error

%s system error description

User Response: Refer to the system error description for an indication of the resolution.

SNX4103E csokcntx(%d): RFCONNECT only AF_INET address types are supported

Description: While processing a CONNECT request, the returned IP address for the remote NetEx hostname is not an IPv4 address.

%d uniquely identifies the internal session ID in use

User Response: Insure the remote NetEx host names being used are resolved correctly.

SNX4104E csokcntx(%d): bequeathsocket failed: %s

Description: On a Unisys system, relinquishing ownership of a socket failed while processing a CONNECT or OFFER.

%d uniquely identifies the internal session ID in use

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4108E csokcntx(%d): clean1() send failed errno=%d1: %s

Description: On a Unix or Unisys system, the send on the internal socket connection failed.

%d uniquely identifies the internal session ID in use

%d1 system error

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4109E csokcntx(%d): clean1() recv failed errno=%d1: %s

Description: On a Unix or Unisys system, the receive on the internal socket connection failed.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4110E csokcntx(%d): clean1(): thread cancel failed errno=%d1: %s

Description: On a Unix system, a pthread cancel event failed.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4115E csokcntx(%d): RFOFFER doportmap LICCHK timed out

Description: Connect to local SNXMAP timeout (no response within 5s).

%d	uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4116E csokcntx(%d): RFOFFER doportmap LICCHK socket error, errno=%d1: %s

Description: Connect to local SNXMAP timeout (no response within 5s).

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4117E csokcntx(%d): RFCONNECT: pipi_init error rc=%d1

Description: PIPI environment initialization failed (IBM only).

%d	uniquely identifies the internal session ID in use
%d1	system status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4118E csokcntx(%d): RFOFFER: pipi_init error rc=%d1

Description: PIPI environment initialization failed (IBM only).

%d	uniquely identifies the internal session ID in use

%d1 system status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4119E csokcntx(%d): RFCONNECT get cctable failed

Description: User-provided code conversion table initialization failed.

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4120E csokcntx(%d): RFOFFER get cctable failed

Description: User-provided code conversion table initialization failed.

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4121E csokcntx(%d): CCTABLE='%s' open failed: %s1

Description: Missing or unable to access code conversion table file.

%d uniquely identifies the internal session ID in use

%s code conversion file

%s1 system error description

User Response: Verify SNXMAP configuration for CCTABLE is correct and that all users have read access to the code conversion file.

SNX4122E csokcntx(%d): getcctable: Invalid table name '%s' at line %d1

Description: Invalid or unknown code conversion table name.

%d uniquely identifies the internal session ID in use

%s code conversion table name

%d1 line number in code conversion table file where error occurred

User Response: Verify code conversion table file contains correct table name.

SNX4123E csokcntx(%d): getcctable: Invalid data or unexpected EOF at line %d1

Description: Code conversion table file is incomplete.

%d uniquely identifies the internal session ID in use

%d1 line number in code conversion table file where error occurred

User Response: Verify code conversion table file contains correct and complete data.

SNX4124E csokcntx(%d): socketpair failed: %s

Description: The IPC channel used by a NetEx session could not be created.

%d uniquely identifies the internal session ID in use

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4201E thread(%d): inheritsocket failed: %s

Description: On a Unisys system, assuming ownership of a socket failed while processing a CONNECT or OFFER.

%d uniquely identifies the internal session ID in use

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4202E thread(%d): scb->scbreq_q is NULL

Description: The thread signaled a request was ready, but no request was sent.

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4203E thread(%d): RFCONNECT pipi_init error rc=%d

Description: On an IBM system, while processing a connect, the PIPI environment initialization failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4204E thread(%d): RFCONNECT socket error, errno=%d1, stat=%d2: %s

Description: While processing a Connect request, the connect to the remote offer socket received an error.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4205E thread(%d): SSL setsockopt error, errno=%d1, stat=%d2: %s

Description: On a Unisys only system, the set secure socket options failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4206E thread(%d): RFCONNECT keepalive setsockopt error

Description: While processing a Connect, the keepalive set socket option failed (connect/client).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4207E thread(%d): RFCONNECT connect error, errno=%d1, stat=%d2: %s

Description: While processing a Connect, the connect to offer on remote host failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4208E thread(%d): RFCONNECT getsockname error, errno=%d1, stat=%d2: %s

Description: While processing a Connect, could not get the local address.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4209E thread(%d): RFCONNECT setup_mvsgsk_env error, errno=%d1

Description: On IBM systems, while processing a Connect, received a system error when initializing the environment.

%d uniquely identifies the internal session ID in use

%d1 system error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4210E thread(%d): RFCONNECT setup_mvsgsk_env error, stat=%d1

Description: On IBM systems, while processing a Connect, environment initialization failed.

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4211E thread(%d): RFCONNECT setup_mvsgsk_soc error, errno=%d1

Description: On IBM systems, while processing a Connect, socket initialization failed with a system error.

%d uniquely identifies the internal session ID in use

%d1 system error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4212E thread(%d): RFCONNECT setup_mvsgsk_soc error stat=%d1

Description: On IBM systems, while processing a Connect, socket initialization failed.

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4213E thread(%d): RFCONNECT find_cert_data rc=%d1

Description: On IBM systems, while processing a Connect, unable to find peer certificate common name.

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4214E thread(%d): RFCONNECT ssl_init error, rc=%d1

Description: On Unix systems, while processing a CONNECT, an SSL initialization error was detected (connect/client).

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4215E thread(%d): RFCONNECT SSL_new failed

Description: On Unix systems, while processing a CONNECT, an SSL session allocation error was received (connect/client).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4216E thread(%d): RFCONNECT SSL_set_fd failed

Description: On Unix systems, while processing a CONNECT, an SSL session socket I/O assignment error was detected (connect/client).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4217E thread(%d): RFCONNECT SSL_connect error, rc=%d1

Description: On Unix systems, while processing a CONNECT, an SSL session error was detected (connect/client).

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4218E thread(%d): RFCONNECT SSL_host check failed

Description: While processing a CONNECT, an SSL peer certificate Common Name error was detected (connect/client).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4219E thread(%d): RFOFFER pipi_init error rc=%d1

Description: On IBM systems, while processing an Offer, the PIPI environment initialization failed.

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4220E thread(%d): BIND error, errno=%d1, stat=%d2: %s

Description: While processing an Offer, the bind for the offer port failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4221E thread(%d): SSL setsockopt error, errno=%d1, stat=%d2: %s

Description: On Unisys systems, while processing an Offer, the set secure connection failed (offer/server).

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4222E thread(%d): listen failed, errno=%d1, stat=%d2: %s

Description: The listen on the offer port failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4223E thread(%d): RFOFFER select on portmapsock error, errno=%d1, stat=%d2: %s

Description: While processing an Offer the select on SNXMAP connection failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4224E thread(%d): RFOFFER select error, errno=%d1, stat=%d2: %s

Description: While processing an Offer the wait for the remote host connection failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4225E thread(%d): RFOFFER accept error, errno=%d1, stat=%d2: %s

Description: While processing an OFFER, the ACCEPT for the remote host connection failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%d2 status

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4226E thread(%d): RFOFFER keepalive setsockopt error

Description: While processing an Offer the set keepalive socket option failed (offer/server).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4227E thread(%d): RFOFFER setup_mvsgsk_env error errno=%d1

Description: On IBM systems, while processing an Offer the GSK environment initialization failed (offer/server).

%d uniquely identifies the internal session ID in use

%d1 system error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4228E thread(%d): RFOFFER setup_mvsgsk_env error stat=%d1

Description: On IBM systems, while processing an Offer the GSK environment initialization failed.

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4229E thread(%d): RFOFFER setup_mvsgsk_soc error, errno=%d1

Description: On IBM systems, while processing an Offer the socket initialization failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4230E thread(%d): RFOFFER setup_mvsgsk_soc error, stat=%d1

Description: On IBM systems, while processing an Offer the GSK socket initialization failed.

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4232E thread(%d): RFOFFER ssl_init error, rc=%d1

Description: On Unix systems, while processing an Offer an SSL initialization error was detected (offer/server).

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4233E thread(%d): RFOFFER SSL_new failed

Description: On Unix systems, while processing an Offer an SSL session allocation error was detected (offer/server).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4234E thread(%d): RFOFFER SSL_set_fd failed

Description: On Unix systems, while processing an Offer an SSL session socket I/O assignment error was detected (offer/server).

%d uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4235E thread(%d): RFOFFER SSL_accept error, rc=%d1

Description: On Unix systems, while processing an Offer an SSL session accept error was detected (offer/server).

%d uniquely identifies the internal session ID in use

%d1 status

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4236E thread(%d): setsockopt TCP_KEEPIDLE failed, errno=%d1: %s

Description: On a Unix system, the set keepalive idle time socket option failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4237E thread(%d): setsockopt TCP_KEEPCNT failed, errno=%d1: %s

Description: On a Unix system, the set keepalive probe count socket option failed.

%d uniquely identifies the internal session ID in use

%d1 system error returned

%s system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4238E thread(%d): setsockopt TCP_KEEPINTVL failed, errno=%d1: %s

Description: On a Unix system, the set keepalive probe interval time socket option failed.

%d	uniquely identifies the internal session ID in use
%d1	system error returned
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4239E thread(%d): setsockopt SO_KEEPALIVE failed, errno=%d1: %s

Description: The set keepalive socket option failed.

%d	uniquely identifies the internal session ID in use
%d1	system error returned
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4240E thread(%d): RFOFFER socket error, errno=%d1, stat=%d2: %s

Description: Offer socket allocation error.

%d	uniquely identifies the internal session ID in use
%d1	system error returned
%d2	status
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4241E thread(%d): RFOFFER bind to addr=%s:%d1 errno 111 %d2 times, max is 5

Description: On an IBM system, bind for offer port failed with errno 111 more than 5 times.

%d	uniquely identifies the internal session ID in use
%s	bind address
%d1	bind port
%d2	number of bind attempts

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4242E thread(%d): RFOFFER no port available in range (%d1-%d2)

Description: Configured port range contains no open ports for offers.

%d	uniquely identifies the internal session ID in use
%d1	first port number
%d2	last port number

User Response: Increase PORTNUM or use ephemeral ports (PORTSTART=0).

SNX4243E thread(%d): RFOFFER getsockname error, errno=%d1, stat=%d2: %s

Description: Error during OFFER on getsockname.

%d	uniquely identifies the internal session ID in use
%d1	system error
%d2	status
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4244E thread(%d): RFCONNECT select error, errno=%d1, stat=%d2: %s

Description: Failure while waiting for Connect to Offer on remote host.

%d	uniquely identifies the internal session ID in use
%d1	system error
%d2	status
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4245E thread(%d): RFCONNECT connect timeout

Description: Connect to Offer on remote host timed out after CONNTO seconds.

%d	uniquely identifies the internal session ID in use

User Response: Verify the remote NetEx hostname is correct or the network path to the host is available. Increase the value of CONNTO.

SNX4301E %s

Description: On Unix systems, an OpenSSL error was detected.

%s The error strings will have the following format:

[pid]:error:[error code]:[library name]:[function name]:[reason string]:[file name]:[line]:[optional text message]

Where:

"pid" and "error code" are 8-digit hexadecimal number

"library name", "function name", and "reason string" are ASCII text, as is "optional text message" if one was set for the respective error code. If there is no text string registered for the given error code, the error string will contain the numeric code.

User Response: Contact support@netex.com for details on the error. Review the previous error messages. FIPS mode is not available and should be disabled in the SNXMAP configuration (FIPSMODE OFF).

SNX4302E Certificate error %d at depth %d1: %s

Description: A local certificate verify error was detected.

%d error id

%d1 certificate depth

%s error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4303E Certificate error %d at depth %d1: %s

Description: During an SSL handshake a peer certificate verify error was detected.

%d error id

%d1 certificate depth

%s error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4304E No certificate file
Local certificate file not specified (see SNXMAP configuration).

Description: Local certificate file not specified.

User Response: Check the SNXMAP configuration file for proper specification.

SNX4305E BIO_new failed

Description: During the local certificate verify, a memory allocation error was detected.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4306E BIO_read_filename failed

Description: During the local certificate verify, an error was detected reading the certificate file.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4307E PEM_read_bio failed

Description: During the local certificate verify, a PEM certificate data read error was detected.

User Response: Check certificate file for valid PEM format.

SNX4308E SSL_CTX_get_cert_store failed

Description: During the local certificate verify, an SSL context certificate store error was detected.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4309E X509_STORE_CTX_new failed

Description: During the local certificate verify, a memory allocation error was detected.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4310E X509_STORE_CTX_init failed

Description: During the local certificate verify, a store initialization error was detected.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4311E ssl_init: init_locks() failed

Description: Failed to initialize the SSL thread-safe lock controls.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4312E ssl_init: SSL_CTX_new failed

Description: Failed to allocate the SSL context.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4313E ssl_init: SSL_CTX_set_cipher_list failed

Description: Failed to set the SSL supported cipher list.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4314E ssl_init: Failed to load trusted CA certificates
SNX4314C ssl_init: Check CAFILE and/or CAPATH are correct

Description: Failed to set the locations for the trusted CA certificates.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4315E ssl_init: Certificate verification failed (%s)

Description: Verify of local certificate failed.

%s certificate file

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4316E ssl_init: SSL_CTX_use_certificate_file (%s) error

Description: Failed to load certificate file.

%s certificate file

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4317E ssl_init: SSL_CTX_use_PrivateKey_file (%s) error

Description: Failed to load private key file.

%s key file

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4318E ssl_init: Private key does not match the certificate

Description: Certificate and private key are inconsistent.

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4319E Get peer address failed: %s

Description: SSL CN/SAN peer certificate check could not get peer address.

%s system-error-description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4320E Convert peer address failed: %s

Description: SSL CN/SAN peer certificate check could not convert peer address.

%s system-error-description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4321E ssl_init: FIPS_mode_set failed

Description: Failed to set FIPS mode.

User Response: Contact your system administrator with details on the error.

SNX4401E gsksetup(%d): req %d1 gsk_environment_open error, rc=%d2

Description: On IBM systems, an error was detected when opening the GSK environment.

%d	uniquely identifies the internal session ID in use
%d1	1: connect, 129: offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4402E gsksetup(%d): req %d1 gsk_attribute_set_enum PROTOCOL_SSLV2 ON error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the protocol to SSL V2.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4403E gsksetup(%d): req %d1 gsk_attribute_set_enum PROTOCOL_SSLV3 ON error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the protocol to SSL V3.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4404E gsksetup(%d): req %d1 gsk_attribute_set_enum PROTOCOL_TLSV1 ON error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the protocol to TLS V1.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4405E gsksetup(%d): req %d1 gsk_attribute_set_buffer KEYRING_FILE '%s' error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the keyring filename.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4406E gsksetup(%d): req %d1 gsk_attribute_set_buffer KEYRING_PW '%s' error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the keyring password.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%s	the keyring password that is being used
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4407E gsksetup(%d): req %d1 gsk_environment_init error, rc=%d2

Description: On IBM systems, an error was detected when attempting initialize the GSK environment.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4408E gsksetup(%d): req %d1 gsk_secure_socket_open error, rc=%d2

Description: On IBM systems, an error was detected when attempting to open the socket.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4409E gsksetup(%d): req %d1 gsk_attribute_set_numeric_value GSK_FD (sock %d2) error, rc=%d3

Description: On IBM systems, an error was detected when attempting to associate the socket with GSK.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	the socket number to be associated with the connection
%d3	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4410E gsksetup(%d): req %d1 gsk_attribute_set_buffer KEYRING_LABEL '%s' error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the keyring label.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%s	the keyring label for the certificate to be used
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4411E gsksetup(%d): req %d1 gsk_attribute_set_enum SESSION_TYPE SERVER_SESSION error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the session type to a server session.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4412E gsksetup(%d): req %d1 gsk_attribute_set_enum SESSION_TYPE CLIENT_SESSION error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the session to a client session.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4413E gsksetup(%d): req %d1 gsk_attribute_set_buffer V2_CIPHER_SPECS NULL error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the connection ciphers.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4414E gsksetup(%d): req %d1 gsk_attribute_set_buffer V3_CIPHER_SPECS NULL error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the connection ciphers.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4415E gsksetup(%d): req %d1 gsk_attribute_set_buffer V3_CIPHER_SPECS_EXPANDED '0035002F000A' error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the connection ciphers.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4416E gsksetup(%d): req %d1 gsk_attribute_set_enum V3_CIPHERS V3_CIPHERS_CHAR4 error, rc=%d2

Description: On IBM systems, an error was detected when attempting to set the connection ciphers

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4417E gsksetup(%d): req %d1 gsk_secure_socket_init error, rc=%d2

Description: On IBM systems, an error was detected when attempting to initialize the secure socket.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4418E gsksetup(%d): req %d1 gsk_attribute_set_enum CLIENT_AUTH_TYPE_ CLIENT_AUTH_FULL_TYPE error, rc=%d2

Description: On IBM systems, an error was detected when attempting to initialize attribute of a secure socket.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4419E gsksetup(%d): req %d1 gsk_attribute_set_enum CLIENT_AUTH_ALERT_CLIENT_AUTH_NOCERT_ALERT_ON error, rc=%d2

Description: On IBM systems, an error was detected when attempting to initialize attribute of a secure socket.

%d	uniquely identifies the internal session ID in use
%d1	1:connect, 129:offer
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4420E pipi_sub_call(%d): CEEPIPI_call_sub for func %d1 (%s) failed - rc is %d2

Description: On IBM systems, an error was detected when attempting to make the CEEPIPI call.

%d	uniquely identifies the internal session ID in us
%d1	the attempted socket function number
%s	the attempted socket function name
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4421E pipi_sub_call(%d): CEEPIPI_call_sub for func %d1 (%s) subretc < 0, errno=%d2

Description: On IBM systems, an error was detected when attempting to make the CEEPIPI call.

%d	uniquely identifies the internal session ID in use
%d1	the attempted socket function number
%s	the attempted socket function name
%d2	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4422E pipi_init(%d): calloc of gskcall_parms failed

Description: On IBM systems, an error was detected when attempting to allocate the CEEPIPI call control block.

%d	uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4423E pipi_init(%d): Out of memory allocating pre-init table

Description: On IBM systems, out of memory was detected when attempting to allocate the pre-init table.

%d	uniquely identifies the internal session ID in use

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4424E pipi_init(%d): __CEEPIPI_init_sub() failed - rc is %d1

Description: On IBM systems, out of memory was detected when attempting to initialize the CEEPIPI interface.

%d	uniquely identifies the internal session ID in use
%d1	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4425E pipi_term(%d): __CEEPIPI_term() failed - rc is %d1

Description: On IBM systems, out of memory was detected when attempting to terminate the CEEPIPI interface.

%d	uniquely identifies the internal session ID in use
%d1	error returned

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4426E pipi_sub_call(%d): __CEEPIPI_call_sub for func %d1 (%s) subretc=%d2 (%s1)

Description: On IBM systems, out of memory was detected when attempting to terminate the CEEPIPI interface.

%d	uniquely identifies the internal session ID in use
%d1	function number
%s	function name
%d2	error returned
%s1	error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4428E gsksetup(%d): req %d1 gsk_fips_state_set %d2(%d3) error, rc=%d4

Description: On IBM systems, set of GSK FIPS state failed.

%d	uniquely identifies the internal session ID in use
%d1	function number
%d2	FIPs level to set
%d3	GSK FIPs level enum
%d4	error number

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4501E doportmap(%d): socket error, errno=%d1: %s

Description: Socket allocation failed.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4502E doportmap(%d): connect error, errno=%d1: %s

Description: SNXMAP connect failure.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4503E doportmap(%d): send request error, errno=%d1: %s

Description: SNXMAP send failure.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4504E doportmap(%d): read response select error, errno=%d1: %s

Description: SNXMAP select failure.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4505E doportmap(%d): recv response error, errno=%d1: %s

Description: SNXMAP receive failure.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4506E doportmap(%d): could not allocate %d1 bytes

Description: On behalf of the CONNECT or OFFER, an out of memory condition is detected by the local SNXMAP facility.

%d	uniquely identifies the internal session ID in use
%d1	amount of data required on the allocation

User Response: Retry the program; system resources may have freed up.

SNX4507E doportmap(%d): connect select error, errno=%d1: %s

Description: Failure while waiting for Connect to remote SNXMAP.

%d	uniquely identifies the internal session ID in use
%d1	system error
%s	system error description

User Response: Contact support@netex.com for details on the error. Review the previous error messages.

SNX4901E TCPCFG errors

Description: While processing the TCPCFG file on Unisys errors were encountered. The errors will be logged following this message.

User Response: Correct the parameters in error

SNX4902E SNXMAP ERROR sending to Console %o

Description: An error occurred when attempting to write to the console. %0 is the octal code. The is in response to an Exec Request 010.

User Response: Currently TERM is the only valid command

SNX4903E KEYIN (%s) FAILED STATUS O-%o FLAG O-%o

Description: A system error occurred while registering the keyin with the system.

User Response: Is the Unisys ER reference manual for the keyin ER.

SNX4904E KEYIN INPUT FAILED -O %o

Description: An error occurred while attempting to register a console keyin.

%0 is the error code from er 0244.

User Response: Verify the KEYIN value is not in use by another application, or the KEYIN value does not start with an alphabetic character, or the length is less than 9.

Appendix C: H214 for z Series/zOS Installation

Prerequisites

The following are hardware and software prerequisites for installing the H214 product.

- A z Series compatible system running a supported zOS release. Review the website for supported OS distributions.
- At least one other processor on the network running Secure NetEx/IP software. This processor should be connected with another Secure NetEx/IP (not required for intra-host test/evaluation).
- Customers must obtain a software KEY from Support@netex.com prior to running the H214 software. Customers must contact Support@netex.com with the customer site name, hostname, and the Secure NetEx/IP product designator (e.g., H214). Support@netex.com will supply the necessary key once this information has been received. The customer needs to place this key into the NESikeys file as discussed later in this section.
- (Required only to use secure job and/or data transfers) Use the gskkyman utility in order to create a key database which must contain the required client and server certificates and any necessary remote server certificates. The name of the key database, its password, and the client and server certificate labels must be specified in the SNXMAP configuration file.

Users need access to OMVS and the key database because of the SSL implementation. Started tasks need access to Unix services and if the jobs have a REGION specified it needs to be at least 1.2M. All requirements for the equipment listed above must be met before proceeding with the installation.

Hardware Installation

Install and verify proper operation of the appropriate operating system.

Accessing the H214 software distribution

The H214 Secure NetEx/IP software is available as a binary XMIT file which may be downloaded from NetEx. Contact Support@netex.com to request the download link.

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 Secure NetEx/IP will be used, and on the particular features supported by H214, and will authorize Secure NetEx/IP to be used on a particular LPAR.

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

Issue the following command on the z/OS system on which Secure 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 H214 will be executed.

An example of the D M=CPU command is shown in Figure 17.

```
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.02.00000003BC7F
CPC SI = 2096.C02.IBM.02.000000000003BC7F
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 17. 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.

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

Telephone: (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 (H214)

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 H214 NetEx/IP Release 1.0 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.

1) Obtain the H214 distribution file.

2) Upload the distribution file to z/OS.

3) TSO RECEIVE the distribution file.

4) Modify and Submit the SNXINST job on z/OS.

5) Check for required updates.

6) Obtain the H214 software key.

7) Review the H214 initialization parameters and Run Install Job

9) Define SNETEX Service

10) (Optional) Update Policy Agent

11) (Optional) System Performance Consideration

12) Create Secure NetEx/IP addressing information (optional)

13) Create Code Conversion Table (optional)

14) Review Installed JCL

15) Start SNXMAP

16) (Optional) Submitting SNXMAPOP for z/OS

18) (Optional) Execute the SNXMVEAT Program

19) (Optional) Execute the SNXMVGEN Program

1) Obtain the H214 distribution file.

The distribution file 'h214vvvv.xmit' can be downloaded from Network Executive Software. Contact support@netex.com to obtain the download instructions.

2) Upload the distribution file to z/OS.

FTP (binary mode) the h214vvvv.xmit file to the z/OS system as follows:

Connect via FTP to your z/OS system.

Change the directory to your desired high-level qualifier:

cd 'high-level-qualifier'

If necessary, change the location of your local directory to the location of the distribution file:

lcd 'directory-name'

Set the required attributes for the file:

quote site lrecl=80 blksize=3120 recfm=fb prim=5000 sec=200 blocks

Transfer the distribution file in bin mode:

bin

put h214vvvv.xmit distpkg.xmit

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'

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 "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 three members:

COMPDD

SNXINST

SNX010

SNXINST is a job that uses SNX010 to create the H214 libraries and performs the installation.

4) Modify and Submit the SNXINST job on z/OS.

The SNXINST installation job consists of three phases:

LOAD Allocates the H214 base installation datasets and loads the datasets from the downloaded distribution.

EDIT Automatically tailors the H214 SNXMAP startup proc and startup JCL.

COPY Copies the tailored SNXMAP startup proc to the specified PROCLIB library.

Tailor the installation job in SNXINST.

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

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.

Review and tailor the installation parameters contained in SNXINST. At a minimum, the following parameters should be specified:

START(BEGIN)

STOP(END)

HLQ(hlqname)

DFILE(dfilename)

Submit the SNXINST job. This job will allocate and load the following H214 distribution libraries:

hlq.SNXCTL	Base control library
hlq.SNXMAC	Base macro library (limited distribution)
hlq.SNXOBJ	Base object library (limited distribution)
hlq.SNXLOAD	Base load library
hlq.SNXALOAD	Base load library (for SNXMAP)

SNXINST installation parameters:

HELP Default value: NO

Allowed values: YES | NO

The HELP parameter is used to produce a description of the H214 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 | EDIT | COPY

The START parameter is used to determine which phase the H214 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 | EDIT | COPY | END

The STOP parameter is used to determine which phase the H214 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 COPY phase, code the START and STOP parameters as:

START(COPY)

STOP(COPY)

HLQ Default value: SNX.H2140140

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 SNX.H2140140 is used, which results in all of the datasets identified in step 0 on page 63 being created using names in the following format:

SNX.H2140150.SNXCTL Base control library

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

Example: HLQ(SNX.H2140150)

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 H214 installation data sets before the data set is updated by the installation job. The COMPRESS parameter will only cause a compress of the H214 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)

DFILE Default value: SNX.SNX.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 “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 H214 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 H214 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)

SNXPROC Default value: SNXMAP

Allowed values: Any valid JCL procedure name

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

Example: SNXPROC(SNXMAP)

SNXMAPCF Default value: hlq.SNXCTL(SNXMAPCF)

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

The SNXMAPCF parameter is used to provide the fully qualified name of the SNXMAP initialization file. This name will be added to the NETEX procedure that is built when performing the installation. Example: SNXMAPCF(SNX.H2140100.DISTCTL(SNXMAPCF)

PRODCONF Default value: hlq.SNXCTL(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 H214 release and would like to incorporate their current PRODCONF file into the new NETEX procedure.

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

LICCODES Default value: hlq.SNXCTL(LICCODES)

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

The LICCODES parameter is used to provide the fully qualified name of an existing Secure 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 H214 release and would like to incorporate their current LICCODES file into the new NETEX procedure.

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

SNXSYSCL Default value: A

Allowed values: A-Z, 0-9

The SNXSYSCL parameter is used to provide the JES output class for SYSOUT datasets in the SNETEX procedure.

Example: SNXSYSCL(H)

SYSTCPD Default value: ‘’ (none)

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

The SYSTCPD parameter is used to provide the fully qualified name of an alternate TCP/IP stack for Secure 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 Secure NetEx/IP.

Example: SYSTCPD(TCPIPZ.TCPIP.DATA)

PROCLIB Default value: (None)

Allowed values: " or any valid fully qualified data set name

The PROCLIB parameter is used to provide the fully qualified name of the JCL procedure library data set that will contain the Secure NetEx/IP startup procedure, as specified by the SNXPROC 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:

PROCLIB('')

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 Secure NetEx/IP proc name specified by SNXPROC 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)

5) Check for required updates.

Refer to the H214 Memo-to-Users for instructions on checking for product updates at www.netex.com.

6) Obtain the H214 software key.

Obtain the appropriate H214 software key from Secure NetEx/IP support (www.netex.com), by providing the output of the 'D M=CPU' command (see "Obtaining the Software Key" on page 59). 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 H214 from executing properly.

A sample PRODCONF file is illustrated in Figure 18. A sample LICCODES file is illustrated in Figure 19.

The LICCODES file may contain multiple keys. If H214 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 H214 can be placed in the same LICCODES file. Each instance of H214 will use the first key found that contains its fingerprint.

If there are multiple keys for the same LPAR, H214 will use the first key found for the LPAR on which H214 is being started, as it sequentially reads the keys from the file. This makes it important to add new H214 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 H214, 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 NetEx eFT213 product is installed, it can also share the same PRODCONF and LICCODES files with H214.

The LICCODES files may also contain comments, as shown in Figure 19.

A PRODCONF DD statement must be included in the H214 startup JCL. If the DDN format is used for the LICPATH specification, then a LICCODES DD statement must also be included in the H214 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 H214 startup JCL. However, in both cases, the LICCODES file must exist, and be able to be allocated and checked during the execution of H214.

Sample JCL to load the PRODCONF and LICCODES files is contained in the "hlq.SNXCTL" 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 18. Sample PRODCONF records

```
*
* H214 Software license key for CPU/LPAR
*
*   Comments (preceded by either an '*' or a '#')
*   may be placed in the LICCODES file.
*
* add software license key obtained from NESi after this statement
*
*   NTX214IP
B6YL-6AEA-BAMF-UAH7-AH7Q-CAPU-GAHS-NYX5-5DZE-ZB26
```

Figure 19. Sample LICCODES record

7) Review the H214 initialization parameters and Run Install Job

A sample configuration file is located in hlq.SNXCTL(SNXMAPCF). Refer to Appendix G for a description of the available statements.

After the installation, member SNXMAPCF should be in the data set described by the SNXMAPCF DD statement in the NetEx start-up JCL. Edit member SNXMAPCF to your site specifications.

8) (Optional) Authorize the SNXALOAD load library.

If using the optional SNXCONS program, use your site procedures to ensure the hlq.SNXALOAD 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 SNXALOAD 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 SNXALOAD library.

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

9) Define SNETEX Service

SNETEX must be defined as a TCP service in TCPIP.ETC.SERVICES

```
SNETEX        6951/tcp        snetex  # secure netex/ip
```

10) (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.

11) (Optional) System Performance Consideration

Consider updating the SCHED member in your parmlib to run the snxmap program as noswappable.

PPT PGMNAME(SNXMAP) NOSWAP

Consider your Workload Manager configuration to insure the SNXMAP task will run at a dispatch level that is appropriate for your system. This is a server application and should run at a server level priority. If you experience NRBSTATs of 3980 while communicating with the SNXMAP program, an internal TCP request did not complete in 10 seconds. This is usually a configuration issue, or the system is very busy and could not service the request.

12) Create Secure NetEx/IP addressing information (optional)

There are two methods of creating/updating the IP information on your system or network to allow for Secure NetEx/IP to operate properly.

1. **Use system name services**

   This method utilizes local system name services (ex. DNS, hosts file…). The SNXMAP configuration parameter USEDNS controls the use of this method.

   If you want to isolate the Secure NetEx/IP hosts to specific IP addresses, the IP hostnames **must** be in the following format (case is important):

   NTX***IPHostname***

   Where ***IPHostname*** is the name of the host to be used.

2. **Use SNXMAP**

   This method utilizes the SNXMAP host table. The SNXMAP configuration parameter USESETIP controls the use of this method.

   Updating the SNXMAP hosts table can be done on demand with the snxmapop setip command or permanently with setip commands in the SNXMAP command file.

13) Create Code Conversion Table (optional)

If you have a saved code conversion table go to next step.

Create a new code conversion table file by copying the example file. The example file contains the contents of the internal conversion tables and is located at ‘hlq.snxctl(CCTABLES)’. Do not use the example file directly as an upgrade may discard the changes.

Edit the table data in the newly created file. Unused tables can be deleted.

Set the value of the CCTABLE directive in the SNXMAP configuration file to the new file. Example:
CCTABLE //DSN:hlq(cctable)

14) Review Installed JCL

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

15) Start SNXMAP

Start SNXMAP by issuing the 'S SNXMAP' command.

Since SNXMAP functions as a long running task, Network Executive Software advises running SNXMAP 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.

Start SNXMAP by entering the following command:

```
S SNXMAP
```

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

16) (Optional) Submitting SNXMAPOP for z/OS

Edit the snxmapop member in file hlq.SNXCTL that was created as part of the installation process.

```
//SNXMAPOP JOB ,SNXMAPOP,CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*        THE FOLLOWING JCL WILL EXECUTE THE SNXMAPOP PGM
//*
//*        -h, --help : program usage
//*        -d : enable debug output
//*        a[ll] : show all registered offers (default)
//*        k[ey] : show license key
//*        l[oad] : load license key from key file
//*        p[arms] : show local config and license status
//*        setc[onip] addr: set connect allow ip addr
//*        seti[p] host addr: set host/ip mapping
//*        showc[onip] : show connect allow ip addrs
//*        showi[p] : show all host/ip mappings
//*        unseti[p] host addr: unset host/ip mapping
//*        unsetc[onip] addr: unset connect allow ip addrs
//*        dbgon : enable debug msgs
//*        dbgoff : disble debug msgs
//*
//*
//SNXMAPOP EXEC PGM=SNXMAPOP,REGION=4096K,
//             PARM='all'
//*            PARM='-d,all'
//*            PARM='key'
//*            PARM='load'
//*            PARM='parms'
//*            PARM='-d,setconip,10.1.1.5'
//*            PARM='-d,setip,zos,10.1.5.156'
//*            PARM='showconip'
//*            PARM='showip'
//*            PARM='unsetconip,10.1.6.225'
//* I          PARM='unsetip,fred,10.1.6.225'
//*            PARM='dbgon'
//*            PARM='dbgoff'
//*            PARM='-h'
//*            PARM='--help'
//*
//STEPLIB  DD  DISP=SHR,DSN=hlq.SNXLOAD
//*
//STDOUT   DD  SYSOUT=*,DCB=(RECFM=FB,LRECL=256,BLKSIZE=256)
//STDERR   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
```

Uncomment the PARM you would like to run and submit the job.

This example shows PARM=’all’ as being uncommented and will show all registered offers (default)

Once you have selected a PARM save the file and submit the job.

NOTE: only 1 PARM can be uncommented at a time.

17) (Optional) Submitting SNXCONS for z/OS

Submit the SNXCONS member in file hlq.SNXCTL that was created as part of the installation process. This program accepts the modify console commands. Any command supported by the SNXMAPOP program can be input to the modify command (/F jobname,command).

Example;

/F SNXCONS,SHOWIP

```
F SNXCONS,SHOWIP
+SHOWIP
+connecting to snxmap at 127.0.0.1:6951
+snxmap status=0
+snxmap hosts=7
+Name     IP Address
+-------- ---------------
+ABCD    208.45.168.74
+ZOS      10.100.0.91
+AIX       10.1.5.101
```

18) (Optional) Execute the SNXMVEAT Program

The SNXMVEAT program "offers" its services. This program works in tandem with SMVGEN.

Execute the SNXMVEAT program. Edit member SNXMVEAT, shown in Figure 20 on page 73, in the "hlq.SNXCTL" data set, and change the parameters to your specifications. The "offer name" for both the SNXMVEAT and SNXMVGEN 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.

```
//SNXMVEAT JOB ,NTXMVSET,CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*        THE FOLLOWING JCL WILL EXECUTE THE SECURE NETEX/IP SNXMVEAT PGM
//*
//*        STDIN : NUMSESS SECURE VALIDATE OFFERNAME HOSTNAME
//*
//SNXMVEAT EXEC PGM=SNXMVEAT,REGION=4096K
//*
//STEPLIB  DD  DISP=SHR,DSN=BETATST.H2140100.SNXLOAD
//*
//*MEATDBG  DD  DUMMY
//*SNXALL  DD  DUMMY
//*SNXGTRC DD  DUMMY
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSTERM  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//*
//*  (OPTIONAL) SYSTCPD POINTS TO ALTERNATE TCP/IP STACK
//*
//*SYSTCPD DD  DISP=SHR,
//*SYSTCPD+1   DSN=TCPIP.TCPIP.DATA
//STDIN    DD  *
001 Y Y SNXMVEAT ZOSA
//*
//
```

Figure 20. Sample SNXMVEAT Job, Member 'SNXMVEAT in hlq.SNXCTL

19) (Optional) Execute the SNXMVGEN Program

The SNXMVGEN "connects" to SNXMVEAT and sends a variable number (and size) of data blocks to the SNXMVEAT program.

Execute the SNXMVEAT program. Edit member SNXMVGEN, shown in Figure 21, in the "hlq.SNXCTL" data set, and change the parameters to your specifications. Specify the value of the host name on which the SNXMVEAT program is executing in the PARM field. The "offer name" for both the SNXMVEAT and SNXMVGEN jobs must be the same. The values specified in the PARM field of the SNXMVGEN are shown in Figure 21. Save the changes and submit the job to z/OS for execution. This program ends when the request is complete.

```
//SNXMVGEN JOB ,SNXMVGEN,CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*        THE FOLLOWING JCL WILL EXECUTE THE SECURE NETEX SNXMVGEN PGM.
//*
//* STDIN : NUMSESS NUMBLKS SIZE ODATA LOOPS DMODE SECURE VALIDATE HOST OFFERNM
//*
//SNXMVGEN EXEC PGM=SNXMVGEN,REGION=4096K
//STEPLIB  DD  DISP=SHR,DSN=BETATST.H2140100.SNXLOAD
//*
//*        STDIN SUPPLIES THE NEEDED INPUT
//*        PARAMETERS TO THE SNXMVSGN UTILITY PROGRAM
//*
//*        SSS      - IS THE NUMBER OF SESSIONS
//*        AAAAA    - IS THE NUMBER OF BLOCKS TO TRANSMIT
//*        BBBBB    - IS THE SIZE (IN BYTES) OF EACH BLOCK
//*        OOO      - IS THE SIZE (IN BYTES) OF ODATA IN EACH BLOCK
//*        LLLL     - IS THE NUMBER OF LOOPS.
//*        DDDD     - IS THE AUTO DATAMODE CHARACTER SET.
//*        S        - SECURE (Y or N).
//*        V        - VALIDATE (Y or N).
//*        EEEEEEEE - IS THE HOST NAME.
//*        FFFFFFFF - IS THE OFFER NAME.
//*
//*MGENDBG  DD  DUMMY
//*SNXALL  DD  DUMMY
//*SNXGTRC DD  DUMMY
//STDOUT   DD  SYSOUT=*
//STDERR   DD  SYSOUT=*
//SYSPRINT DD  SYSOUT=*
//SYSTERM  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//*
//*  (OPTIONAL) SYSTCPD POINTS TO ALTERNATE TCP/IP STACK
//*
//*SYSTCPD DD  DISP=SHR,
//*SYSTCPD+1   DSN=TCPIP.TCPIP.DATA
//STDIN    DD  *
001 01000 16000 000 0001 0000 Y N ZOSA SNXMVEAT
//*
//                                                                      //
```

Figure 21. Sample SNXMVGEN Job, Member 'SNXMVGEN in hlq.SNXCTL

Debugging User Applications

Users may wish to see Secure NetEx/IP information when attempting to debug application problems. The following can be used in the application job as DD DUMMY cards. By default, the following diagnostic information will go to STDOUT or STDERR.

SNXERR	display errors seen by Secure NetEx/IP
SNXINF	display informational messages seen by Secure NetEx/IP
SNXDBG	debug messages seen by Secure NetEx/IP
SNXTRC	display the NRBS on input and completion
SNXALL	display all messages except SNXGTRC and SNXLETRC
SNXGTRC	trace the gsk traffic (file gsk.<process>.ff.trace will be put in the z/OS UNIX /tmp directory)
SNXLETRC	trace the LEC calls

Considerations For Applications Using H214

When building a Secure NetEx/IP application, the load modules of "THREAD" and "LEGSK" must be available to the application at run time. This can be accomplished by copying them from the H214 <HLQ>.SNXLOAD" to the library the application will be running from, or by adding a //STEPLIB DD DSN= <HLQ>.SNXLOAD,DISP=SHR to the JCL for your application. When your application is linked, it must specify "INCLUDE DISTOBJ(SNXLIB)" where DISTOBJ is a ddname specifying the <HLQ>.DISTOBJ distribution library. Adding the "SNXINFO DD DUMMY" to JCL will show the version of SNXLIB that your application was linked with.

Whenever a new version of H214 is installed, the Secure NetEx/IP application must be relinked with the new SNXLIB and the “THREAD” and “LEGSK” load modules must be copied again or the //STEPLIB in the application updated.

Appendix D: H304 for Unisys Dorado/OS2200 Installation

Prerequisites

The following are hardware and software prerequisites for installing the H304 product.

- A Unisys Dorado compatible system running a supported OS2200 release. Review the website for supported OS distributions.
- At least one other processor on the network running Secure NetEx/IP software. This processor should be connected with another Secure NetEx/IP (not required for intra-host test/evaluation).Customers must contact Support@netex.com with the customer site name, system-info (see Post Installations Step 1), and the Secure NetEx/IP product designator (e.g., H304). Support@netex.com will supply the necessary key once this information has been received. The customer needs to place this key into the NESikeys file as discussed later in this section.
- H304 can coexist in the same OS2200 partition as H300e and/or H300IPC. They will each need a separate license key, as they are different products.
- Certificates and Key: If your certificates will NOT be signed by a third party, the self-signed certificates will need to be installed on the systems using secure transfer. For a system using CPCOMM, the certificate will need to be copied into the CPCOMM configuration file pointed to by the "TRUSTED_CERTIFICATE_FILE" parameter of the SSLTLS-security statement. For CPCOMM/OS, all certificates installed are trusted.

  For CPCOMM the sample SSL/TLS statement could look like:

```
SSL/TLS-SECURITY,SECCPFTP ;
  RSA-PRIVATE-KEY-FILE,DON*KEY.TESTPR ;
  RSA-CERTIFICATE-FILE,DON*CERT.CERTSIGNED ;
  TRUSTED-CERTIFICATES-FILE,DON*CERT.CERTSIGNED ;
  CIPHER-SUITE-MINIMUM,RSA_WITH_RC4_128_MD5
```

  For CPCOMM/OS the sample SSL/TLS statement could look like:

```
SSL/TLS-SECURITY,SECCPFTP ;
      RSA-PRIVATE-KEY-FILE,UNID4150key.pem ;
      RSA-CERTIFICATE-FILE,UNID4150cert.pem ;
.     These above two files reside in the SAIL partition
      CIPHER-SUITE-MINIMUM,RSA_WITH_NULL_MD5
```

  Care should be used when specifying the "CIPHER-SUITE-MINIMUM". Setting it to high could prevent system from negotiating a connection. Setting it to low, allows for simpler algorithms to be used, allowing for easier cracking of the security. The OS2200 operating system will negotiate to the highest possible level of security with partner system.

  Configure the COMAPI interface into CPCOMM;

  The COMAPI used by the secure Netex should specify a local IP V4 address to use to communicate with other components on the same host. By default, this should be 127.0.0.1. This address must match the

IPv4addr in the TCPCFG element in the load library. The COMAPI parameter in this element must match the COMAPI mode you wish to use.

The PROCESS statement for COMAPI **must include** the following three statements:

SSL/TLS-CLIENT-AUTH,WEAK;
SSL/TLS-SECURITY,SECCPFTP ; Points to the correct SSL/TLS statement
INPUT-QUEUE-THRESHOLDS,50,1000,1000000

IT MUST NOT INCLUDE:

SSL/TLS-SECURE-PORTS,nnnn

Note: The COMAPI interface can support application imposed security or system imposed security. It cannot support both features in the same application. Multiple copies of COMAPI can be configured if both features are required. Secure Transfer allows a connection to the COMAPI interface running in a mode other than "A". (See the COMAPI = configuration parameter).

All requirements for the equipment listed above must be met before proceeding with the installation.

Hardware Installation

Install and verify proper operation of the appropriate operating system.

Accessing the H304 Software Distribution

The H304 Secure NetEx/IP software is available as a CFMT file. This file should be FTPed to your OS2200 system using the BIN, and QUOTE SITE CFMT options in ftp. Contact Support@netex.com to request the download link.

Upgrading H304

This is a new product. You must follow the "Software Installation" instructions.

Removing H304

You will then need to delete any files used by H304.

Software Installation

1. No software installation is required for H304. The release file is distributed as an executable program library. The only programs that can be executed from this library are the SNXMAP, SNXMAPOP, SNXMVEAT, SNXMVGEN and SYSINFO programs. You may run these programs from the release library or copy the entire file to a new program library. This file is used to link with user applications like BFX that utilize the secure NETEX interface.

Post Installation Considerations

Configuring H304

Once the software package installation has been successfully completed, Secure NetEx/IP must be configured prior to execution. Sample elements can all be found in your EXEC file.

SNXMAP/UNIECL	contains sample ECL to start the NetEx portmapper application.
SNXMAPOP/UNIECL	contains sample ECL to start the console interface
SNXMAP/CFG	contains the NetEx initialization parameters.

The SNXMAP/UNIECL, and SNXMAPOP/UNIECL should be copied to your SYSLIB*RUN$. These should be renamed to reflect the run names you wish to use. The RELEASE file should not be updated. Update the copy in your SYSLIB*RUN$ file.

SNXMAP/UNIECL:

Correct the run card to reflect your run name, accounting codes and userid requirements
Change the LOG file ECL to reflect the destination of your portmapper output.
Change the Dump data set to reflect the file name for the pads diagnostic file.
Change the copy statements to reflect your Secure NetEx/IP location.
Change the PROD$CONF to point to the keys file (Step 1).
Change the SNXMAPCFG statement to point to your Secure NetEx/IP configuration data (Step2).

SNXMAPOP/UNIECL :

Correct the run card to reflect your run name, accounting codes and userid requirements
Change the PRINT file ECL to reflect the destination of your console output.

1. Create the 'NESikeys' file if necessary

A single NetEx License Key file must reside on each system where one or more NetEx products containing license support will be installed. The following guidelines apply:

- The SNXMAP/UNIECL use statement (@use PROD$CONF) should point to the product configuration file to use. This file should contain two members. The PRODCONF member should contain a single LICPATH statement naming this library. The second element is KEY. This element contains the product key issued by Support@netex.com. Multiple keys can exist in this file; the product will find the valid key. Sample elements are in the release file and maybe copied in to use a template in creating this file.
- In the KEY element, a leading '#', '*', '"', '/', or '!' character denotes a comment line.
- The systems programmer installing this software must edit this file (in quarter-word format (@ed,uq)) to add a new encrypted Software Key each time such a key is obtained from Support@netex.com for H304 and/or other license-enabled NetEx products. This should be done prior to installing the product and must be done prior to any attempt to run the product successfully.
- To obtain a key, contact Support@netex.com, supplying the fingerprint of the machine the software is to be installed on. This may be obtained by using the following ECL:

  @USE RELEASE.,NETEX*RELEASEFILE. (This was FTP'ed to your system.)

  @XQT RELEASE.SYSINFO

(*This is the same information used in the H300IPC product*.)

- The KEY element may contain multiple keys per product due to new product releases or a change to the platform's fingerprint. If there are multiple keys the product will use the first key found that matches the product name and system fingerprint starting at the top of the file. This makes it important to add new keys for an existing system to the top of the file. For example, if a new key is installed that provides a license date extension, or adds a new feature, adding this new key to the file before the old key ensures the new key will be used rather than the old key. To make the file easier to maintain over time, it is recommended that you precede each Key entry with a comment line that documents the product designator (e.g., H304) and release level of the product that the key is associated with. It will then be easier to delete older Keys that are no longer valid for the product.

- Example of creating the product configuration file.
  - @cat,p netex*prodconf.,///50
  - @copy,s release.prodconf to the file cataloged.
  - @copy,s release.key to the file cataloged
  - The prodconf member, contains the name of the file to use (netex*prodconf). It some other name is used, update the element prodconf to point to the correct file.
  - To update the KEY to your prodconf file

    @ed,uq netex*prodconf.KEY (This file must be in quarter word format)

    Insert a line and paste in the key supplied to you. It must start in column 1.

    Comments may be added.
  - The following shows an example of what a *NESikeys* file might look like after adding a key to the file:

```
# Network Executive Software, Inc. Software License Key file
# Key for H304 R1.0:
CGGZ-4AAA-AAAE-IAO5-O5OJ-SBHX-AUZ5-PL4D
```

Create the SNXMAP configuration file

See "Appendix G: Secure NetEx/IP Configuration" on page 95.

This file is pointed to by the snxmap/uniecl use statement SNXMAPCFG use statement. This contains the configuration parameters for the SNXMAP program. This file contains the Secure NetEx/IP hostname for this host, as well as the operator keyin. (snxmap). Term is the only valid command. Secure NetEx/IP hostnames may duplicate the Netex hostname used by the non-secure versions of the product. A sample member (SNXMAP/CFG) is in the release file.

EXAMPLE:

@cat,p netex*snxmap-cfg.,///1000

@copy,i release. SNXMAP/CFG,netex*snxmap-cfg.

@ed,uq netex*snxmap-cfg.

Insert a new line containing the Secure NetEx/IP host name for this system

LCLHOST NETXNME /* NETXNME is the Secure NetEx/IP Host name */

Edit the TCPCFG file

Users will need to configure the application interface to allow programs like BFX to communicate with TCP. This is located in your load library, or the release file.

- COMAPI
  - This is the mode value for the COMAPI you wish to use. Valid values are A-Z.
- INTV4ADDR
  - This is an internal IP V4 ip address assigned to the configured COMAPI Secure NetEx/IP will be using.
  - This is normally 127.0.0.1. Other 127 addresses could be used for different COMAPIs.
- PTMPORT
  - This is the port number Secure NetEx/IP will communicate with the SNXMAP application. 3919 is the default. This must be a site wide agreed upon port.
- RTLVL
  - This is the realtime priority level to be used. 0 is off, 5 to 35 are installation acceptable settings. 5 is high, 35 is low.

Create Secure NetEx/IP addressing information

There are two methods of creating/updating the IP information on your system or network to allow for Secure NetEx/IP to operate properly.

1. **Use system name services**

   This method utilizes local system name services (ex. DNS, hosts file…). The SNXMAP configuration parameter USEDNS controls the use of this method.

   If you want to isolate the Secure NetEx/IP hosts to specific IP addresses, the IP hostnames **must** be in the following format (case is important):

   NTX***IPHostname***

   Where ***IPHostname*** is the name of the host to be used.

   See CPCOM CONFIGURATION AND OPERATIONS GUIDE

2. **Use SNXMAP**

   This method utilizes the SNXMAP host table. The SNXMAP configuration parameter USESETIP controls the use of this method.

   Updating the SNXMAP hosts table can be done on demand with the snxmapop setip command or permanently with setip commands in the SNXMAP command file.

Create Code Conversion Table (optional)

If you have a saved code conversion table go to next step.

Create a new code conversion table file by copying the example file. The example file contains the contents of the internal conversion tables and is located in the release file. The element is CCTABLE1. Do not use the example file directly as an upgrade may discard the changes.

Edit the table data in the newly created file. Unused tables can be deleted.

Set the value of the CCTABLE directive in the SNXMAP configuration file to the new file. Example: CCTABLE netex*file.element

2. Start SNXMAP

Start the SNXMAP run proc. This is a server application. Set the performance level to insure the correct dispatching priority. If you experience NRBSTATs of 3980 while communicating with the SNXMAP program, an internal TCP request did not complete in 10 seconds. This is usually a configuration issue, or the system is very busy and could not service the request.

3. Start SNXMAPOP

snxmapop can run from a telnet terminal window on your OS2200 system. In this case all input and output is directed to the terminal window only.

```
@xqt H304*release000.snxmapop
```

This can be used as a simple configuration validity test. Pressing enter when prompt for input, the SNXMAPOP program will communicate with the SNXMAP program and display and Offers that are currently outstanding.

Submit SNXMAPOP that was created as part of the installation process. This program accepts the modify console commands. Any command supported by the SNXMAPOP program can be input.

Example:

```
@xqt netex*snetex-zos5r.snxmapop
>showip
connecting to snxmap at 127.0.0.8:6951
snxmap status=0
snxmap hosts=5
Name      IP Address
--------  ---------------
ABCD     208.45.168.74
ZOS       10.100.0.91
AIX       10.1.5.101
>
```

Debugging User Applications

Users may wish to see Secure NetEx/IP information when attempting to debug application problems. In your run stream you may add any of the following statements:

```
@USE SNXERR,tpf$     . display errors seen by Secure NetEx/IP
@USE SNXINF,tpf$           . display informational messages seen by Secure
NetEx/IP
@USE SNXDBG.,tpf$    .  debug messages seen by Secure NetEx/IP
@USE SNXTRC.,tpf$   .display the NRBS on input and completion
@USE SNXALL.,tpf$    .display all messages
@USE COMAPIDBG.,tpf$ .display comapi logging information
```

Appendix E: H804 Linux Installation

Prerequisites

The following are hardware and software prerequisites for installing the H804 product.

- An Intel compatible system running a supported Linux OS. Review the website for supported OS distributions.
- At least one other processor on the network running Secure NetEx/IP software. This processor should be connected with another Secure NetEx/IP (not required for intra-host test/evaluation).
- Customers must obtain a software KEY from Support@netex.com prior to running the H804 software. Customers must contact Support@netex.com with the customer site name, hostname, and the Secure NetEx/IP product designator (e.g., H804). Support@netex.com will supply the necessary key once this information has been received. The customer needs to place this key into the NESikeys file as discussed later in this section.
- Certificate and Key: Contact your Security Administrator to obtain PEM formatted keys and certificates. By default, these files are expected in the following files and their location can be specified in the snxmap.cfg file:

  /usr/share/nesi/snetex/key.pem

  usr/share/nesi/snetex/cert.pem

  If you would like to generate a self-signed certificate for testing, you can utilize OpenSSL tools to do this. Any self-signed certificate will need to be added to the trusted CA store for Secure NetEx/IP in order for the secure connection to work.

  One example of this is the following command:

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout
/usr/share/nesi/snetex/key.pem -out /usr/share/nesi/snetex/cert.pem
```

All requirements for the equipment listed above must be met before proceeding with the installation.

Hardware Installation

Install and verify proper operation of the appropriate operating system.

Accessing the H804 software distribution

The H804 Secure NetEx/IP software is available as an RPM which may be downloaded from NetEx. Contact Support@netex.com to request the download link.

Getting the NetEx Public Key

The RPM software distribution package is signed to ensure integrity and authenticity. It is recommended to install the NetEx public key and verify the signature of any software packages before installation.

You can download the NetEx public key from the H804 document page on the netex.com/support website.

Importing the NetEx Public Key

Install the public key as super user with the command:

```
# rpm --import RPM-GPG-KEY-netex.txt
```

Verifying Signatures

You can verify the RPM signature to ensure that a package has not been modified since it has been signed. Verification will also check that a package is signed by the vendors or packagers key.

To verify the signature, use the -K or –checksig option to the rpm command:

```
# rpm -K H804-1.0.i386.rpm
```

Software Removal

During RPM removal, any customized files and log files (e.g., **snxmap.cfg**, **prodconf, NESikeys,** etc.**)** will not be deleted. Remove the software as super user with the command:

```
# rpm -e H804
```

Software Installation

If this is an initial installation, install the software as super user with the command:

```
# rpm -i H804-1.0.i386.rpm
```

If the NetEx public key has not been installed use the command:

```
# rpm -i --nosignature H804-1.0.i386.rpm
```

Upgrading H804

If you are upgrading an existing installation of H804, it is strongly recommended that any running Secure NetEx/IP process be stopped. *If you are upgrading from a release which is older than the most recent previous release you should save the configuration files, remove or uninstall, and perform the instructions for an initial install.* Using the "rpm -U" command preserves any customized files in this package and the replacement files are installed with extensions of ".rpmnew". Any files that are not in the package but in package directories will also be preserved. Upgrade the software as super user with the command:

```
# rpm -U H804-1.0.i386.rpm
```

If the NESi public key has not been installed use the command:

```
# rpm -U --nosignature H804-1.0.i386.rpm
```

Removing H805 RPM

If H805 BFX was installed, this RPM for BFX may need to be removed first (i.e. if the above upgrade fails due to H805 requiring an older H804).

If you wish to remove the rpm, you may use the following command using your superuser id.

```
# rpm -e H805
```

During RPM removal, any customized files and log files will not be deleted.

Removing the NESi Public Key

To remove the NESi public key, as super user issue the command:

```
# rpm –e gpg-pubkey-3d6b35d3-51bb5907
```

Starting, Stopping & Verifying Install of Secure NetEx/IP

For Unix/Linux systems (SysV Init):

The service command should be used to stop, start or restart snetex:

```
# service snetex stop
# service snetex start
# service snetex restart
```

The chkconfig command should be used to verify installation:

```
# chkconfig --list snetex
```

```
The snetex log can be found in /var/log/snxmap
```

For Unix/Linux systems (Systemd):

The systemctl command should be used to stop, start, restart and verify snetex:

```
# systemctl stop snetex.service
# systemctl start snetex.service
# systemctl restart snetex.service
# systemctl status snetex.service
```

The snetex log is managed by the systemd journal. To view the log use:

```
# journalctl -u snetex
```

Post Installation Considerations

Configuring H804

Once the software package installation has been successfully completed, Secure NetEx/IP must be configured prior to execution. The following instructions address editing the files associated with product activation (**/usr/share/nesi/snetex/prodconf** and **/usr/share/nesi/NESikeys**), the configuration file for Secure NetEx/IP (**/usr/share/nesi/snetex/snxmap.cfg**) and starting the Secure NetEx/IP process.

Configuring the H804IP Secure NetEx/IP software consists of the following steps:

1. Edit the ‘NESikeys’ file
2. Edit the snxmap.cfg file
3. Create Secure NetEx/IP addressing information
4. Start SNXMAP
5. Verify that ‘snxmap’ Starts Automatically On Reboot

Edit the ‘NESikeys’ file

A single NESi License Key file must reside on each system where one or more NESi products containing license support will be installed. An example of this file can be found at /usr/share/nesi/snetex. The following guidelines apply:

- The default file name is *NESikeys*.
- The LICPATH keyword/value pair in the product configuration file (see **/usr/share/nesi/snetex/prodconf**) specifies the full path name to this file. The default is: ***/usr/share/nesi/NESikeys***.
- A leading '#', '*', or ';' character, in a file record denotes a comment line.
- The systems programmer installing this software must edit this file to add a new encrypted Software Key each time such a key is obtained from NESi for H804 and/or other license-enabled NESi products. This should be done prior to installing the product and must be done prior to any attempt to run the product successfully.
- To obtain a key from NESi, contact NESi support, supplying the hostname of the machine the software is to be installed on. The hostname may be obtained by issuing the Linux command "*hostname*" with no parameters.
- The file may contain multiple keys per product due to new product releases or a change to the platform's fingerprint (on UNIX this corresponds to the hostname for the target host). If there are multiple keys the NESi product will use the first key found that matches the product name and system fingerprint starting at the top of the file. This makes it important to add new keys for an existing system to the top of the file. For example, if a new key is installed that provides a license date extension, or adds a new feature, adding this new key to the file before the old key ensures the new key will be used rather than the old key. If there are multiple keys the NESi product will use the first key found that matches the product name and system fingerprint starting at the top of the file. This makes it important to add new keys for an existing system to the top of the file. For example, if a new key is installed that provides a license date extension, or adds a new feature, adding
- this new key to the file before the old key ensures the new key will be used rather than the old key. To make the file easier to maintain over time, it is recommended that you precede each Key entry with a comment line that documents the product designator (e.g., H804) and release level of the product that the key is associated with. It will then be easier to delete older Keys that are no longer valid for the product.
- The following shows an example of what a *NESikeys* file might look like after adding several Keys to the file:

```
# Network Executive Software, Inc. Software License Key file
# Key for H804 R1.0:
CGGZ-4AAA-AAAE-IAO5-O5OJ-SBHX-AUZ5-PL4D
```

Edit the snxmap.cfg file

See Appendix G: Secure NetEx/IP Configuration on page 95.

Create Secure NetEx/IP addressing information

There are two methods of creating/updating the IP information on your system or network to allow for Secure NetEx/IP to operate properly.

1. **Use system name services**

   This method utilizes local system name services (ex. DNS, hosts file…). The SNXMAP configuration parameter USEDNS controls the use of this method.

   If you want to isolate the Secure NetEx/IP hosts to specific IP addresses, the IP hostnames **must** be in the following format (case is important):

   NTX***IPHostname***

Where ***IPHostname*** is the name of the host to be used.

2. **Use SNXMAP**

   This method utilizes the SNXMAP host table. The SNXMAP configuration parameter USESETIP controls the use of this method.

   Updating the SNXMAP hosts table can be done on demand with the snxmapop setip command or permanently with setip commands in the SNXMAP command file.

Create Code Conversion Table (optional)

If you have a saved code conversion table go to next step.

Create a new code conversion table file by copying the example file. The example file contains the contents of the internal conversion tables. Do not use the example file directly as an upgrade may discard the changes. ex. (cd /usr/share/nesi/snetex && cp cctable.example cctable).

Edit the table data in the newly created file. Unused tables can be deleted.

Set the value of the CCTABLE directive in the SNXMAP configuration file to the new file (ex. CCTABLE /usr/share/nesi/snetex/cctable).

Start SNXMAP

On an initial install SNXMAP will not start automatically nor will it start following an update. To Start SNXMAP use the following command:

```
service snetex start
```

Verify that 'snxmap' Starts Automatically On Reboot

SNXMAP has been configured to automatically start for run levels 2, 3, and 5 after a system reboot. It will not work until the 'snxmap.cfg' file has been properly set up and placed in the correct location, and the key is correct and in the correct location.

Note: If SNXMAP is started/stopped manually, the following script should be used as it properly modifies some system parameters required by Secure NetEx/IP and detects common problems:

```
service  snetex  start | stop | restart | status
```

Debugging User Applications

Users may wish to see Secure NetEx/IP information when attempting to debug application problems. To enable messages set any of the following variables in the application runtime environment. No value is necessary, the existence of a variable is sufficient (ex. export SNXTRC=).

SNXERR	Display errors seen by Secure NetEx/IP.
SNXINF	Display informational messages seen by Secure NetEx/IP.
SNXDBG	Display debug messages seen by Secure NetEx/IP.
SNXTRC	Display the NRBs on input and completion.
SNXALL	Display all messages (includes all of the above).
SNXSSLTRC	Display SSL state messages (must be set independently).

Appendix F: H624 AIX Installation

Prerequisites

The following are hardware and software prerequisites for installing the H624 product.

- An IBM Power System or System p server running, AIX® 5.3 to 7.1 distributions.
- At least one other processor on the network running Secure NetEx/IP software. This processor should be connected with another Secure NetEx/IP (not required for intra-host test/evaluation).
- Customers must obtain a software KEY from NESi prior to running the H624 software. Customers must contact NESi customer support with the customer site name, hostname, and the Secure NetEx/IP product designator (e.g., H624). NESi customer support will supply the necessary key once this information has been received. The customer needs to place this key into the NESikeys file as discussed later in this section.
- Certificate and Key: Contact your Security Administrator to obtain PEM formatted keys and certificates. These must be place in the following files:

  /usr/share/nesi/snetex/key.pem

  usr/share/nesi/snetex/cert.pem

  If you would like to generate a self-signed certificate for testing, you can utilize OpenSSL tools to do this. Any self-signed server certificate will need to be added to the trusted CA store for Secure NetEx/IP in order for the secure connection to work.

  One example of this is the following command:

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout
/usr/share/nesi/snetex/key.pem -out /usr/share/nesi/snetex/cert.pem
```

All requirements for the equipment listed above must be met before proceeding with the installation.

Hardware Installation

Install and verify proper operation of the AIX® system.

Accessing the H624 software distribution

The H624 Secure NetEx/IP software is available as an RPM which may be downloaded from NESi. Contact NESi Customer Support to request the download link.

Software Removal

During RPM removal, any customized files and log files (e.g., **snxmap.cfg**, **prodconf, NESikeys,** etc.**)** will not be deleted. Remove the software as super user with the command:

```
# rpm –e H624
```

Software Installation

All installation steps must be completed by a user logged on as root.

Version 1.3 of H624 Secure NetEx/IP installs from an RPM package.

Upgrading H624

If you are upgrading an RPM installation of H624, it is strongly recommended that any running Secure NetEx/IP process be stopped. Using the "rpm -U" command preserves any customized files in this package and the replacement files are installed with extensions of ".rpmnew". Any files that are not in the package but in package directories will also be preserved. Upgrade the software as super user with the command:

```
# rpm –U H624-1.3.ppc.rpm
```

Removing H625 RPM

If H625 BFX was installed, this RPM for BFX may need to be removed first (i.e. if the above upgrade fails due to H625 requiring an older H624).

If you wish to remove the rpm, you may use the following command using your superuser id.

```
# rpm –e H625
```

During RPM removal, any customized files and log files will not be deleted.

Starting, Stopping & Verifying Install of SNXMAP

The startsrc command or SMIT should be used to start SNXMAP:

```
# startsrc -s snetex
```

The stopsrc command or SMIT should be used to stop SNXMAP:

```
# stopsrc -s snetex
```

The lssrc and lsitab commands or SMIT can be used to verify installation:

```
# lssrc -S -s snetex
# lsitab snetex
```

Post Installation Considerations

Configuring H624

Once the software package installation has been successfully completed, Secure NetEx/IP must be configured prior to execution. The following instructions address editing the files associated with product activation (**/usr/share/nesi/snetex/prodconf** and (**/usr/share/nesi/NESikeys**), the configuration file for Secure NetEx/IP (**/usr/share/nesi/snetex/snxmap.cfg**) and starting the Secure NetEx/IP process.

Configuring the H624 Secure NetEx/IP software consists of the following steps:

- Edit the 'NESikeys' file
- Edit the SNXMAP.CFG file
- Create Secure NetEx/IP addressing information
- Starting/Stopping
- Verify that 'snxmap' Starts Automatically On Reboot

Edit the 'NESikeys' file

A single NESi License Key file must reside on each system where one or more NESi products containing license support will be installed. An example of this file can be found at /usr/share/nesi/snetex. The following guidelines apply:

- The default file name is **/usr/share/nesi/*NESikeys***.
- The LICPATH keyword/value pair in the product configuration file (see **/usr/share/nesi/snetex/prodconf**) specifies the full path name to this file. The default is: ***/usr/share/nesi/NESikeys***
- A leading '#', '*', or ';' character, in a file record denotes a comment line.
- The actual key must reside on a single line by itself and start in column 1.
- The systems programmer installing this software must edit this file to add a new encrypted Software Key each time such a key is obtained from NESi for H624 and/or other license-enabled NESi products. This should be done prior to installing the product and must be done prior to any attempt to run the product successfully.
- To obtain a key from NESi, contact NESi support, supplying the hostname of the machine the software is to be installed on. The hostname may be obtained by issuing the Linux/AIX command "*hostname*" with no parameters.
- The file may contain multiple keys per product due to new product releases or a change to the platform's fingerprint (on UNIX this corresponds to the hostname for the target host). If there are multiple keys the NESi product will use the first key found that matches the product name and system fingerprint starting at the top of the file. This makes it important to add new keys for an existing system to the top of the file. For example, if a new key is installed that provides a license date extension, or adds a new feature, adding this new key to the file before the old key ensures the new key will be used rather than the old key. To make the file easier to maintain over time, it is recommended that you precede each Key entry with a comment line that documents the product designator (e.g., H624) and release level of the product that the key is associated with. It will then be easier to delete older Keys that are no longer valid for the product.

- The following shows an example of what a *NESikeys* file might look like after adding a key to the file. The first two lines are comments with the key on the third line. Additional comments and keys could be added.

```
# Network Executive Software, Inc. Software License Key file
# Key for H624 R1.0:
CGGZ-4AAA-AAAE-IAO5-O5OJ-SBHX-AUZ5-PL4D
```

Edit the SNXMAP.CFG file

See Appendix G: Secure NetEx/IP Configuration on page 95.

Create Secure NetEx/IP addressing information

There are two methods of creating/updating the host/IP information on your system or network to allow for Secure NetEx/IP to operate properly.

1. **Use system name services**

   This method utilizes local system name services (ex. DNS, hosts file…). The SNXMAP configuration parameter USEDNS controls the use of this method.

   If you want to isolate the Secure NetEx/IP hosts to specific IP addresses, the IP hostnames **must** be in the following format (case is important):

   NTX***IPHostname***

   Where ***IPHostname*** is the name of the host to be used.

2. **Use SNXMAP**

   This method utilizes the SNXMAP host table. The SNXMAP configuration parameter USESETIP controls the use of this method.

   Updating the SNXMAP hosts table can be done on demand with the snxmapop setip command or permanently with setip commands in the SNXMAP command file.

Create Code Conversion Table (optional)

If you have a saved code conversion table go to next step.

Create a new code conversion table file by copying the example file. The example file contains the contents of the internal conversion tables. Do not use the example file directly as an upgrade may discard the changes. ex. (cd /usr/share/nesi/snetex && cp cctable.example cctable).

Edit the table data in the newly created file. Unused tables can be deleted.

Set the value of the CCTABLE directive in the SNXMAP configuration file to the new file (ex. CCTABLE /usr/share/nesi/snetex/cctable).

Starting/Stopping SNXMAP

On an initial install SNXMAP will not start automatically nor will it start following an update. To Start SNXMAP you may use the SMIT tool on AIX. SNXMAP is defined as an AIX subsystem, and may be started or stopped through the SMIT interface. It may also be manually started or stopped with:

```
startsrc –s snetex
stopsrc –s snetex
```

Verify that ‘snxmap’ Starts Automatically On Reboot

SNXMAP has been configured to automatically start for run level 2 after a system reboot. It will not work until the ‘snxmap.cfg’ file has been properly set up and placed in the correct location, and the key is correct and in the correct location. To permanently disable this, the super user may issue the command “rmitab snetex”.

Debugging User Applications

Users may wish to see Secure NetEx/IP information when attempting to debug application problems. To enable messages set any of the following variables in the application runtime environment. No value is necessary, the existence of a variable is sufficient (ex. export SNXTRC=).

SNXERR	Display errors seen by Secure NetEx/IP.
SNXINF	Display informational messages seen by Secure NetEx/IP.
SNXDBG	Display debug messages seen by Secure NetEx/IP.
SNXTRC	Display the NRBs on input and completion.
SNXALL	Display all messages (includes all of the above).
SNXSSLTRC	Display SSL state messages (must be set independently).

Appendix G: Secure NetEx/IP Configuration File

Edit the snxmap.cfg file

The *snxmap.cfg* file contains default values for Secure NetEx/IP parameters.

Note for UNIX only: After an update, if a snxmap.cfg.rpmnew exists there may be updated defaults that should be reviewed.

Edit this file with the following recommendations:

1. The LCLHOST defines the name of your local host. (This is analogous to the NetEx hostname in the NCT of legacy NetEx products.)

2. Edit or modify any other parameters for your host and site. The following table lists all of the parameters and their default values.

3. Certificates and CAs must be configured for secure transfers to operate. Secure NetEx/IP implements client and server certificate verification. Server certificate will be verified by the client and the client certificate will be verified by the server. Certificates and/or CAs will need to be configured on all hosts utilizing a secure connection.

Lines proceeded by an '*', '#', '"', '!', '.'or '/' are comments. In the distributed file, these comments indicate the use of provided program defaults. To override these default values, you should duplicate the entry and uncomment, remove the *, and supply your override value. All keywords and Boolean values are case-insensitive. Boolean values: on, off, yes, no, true, false, 1, 0. The parameters are defined in four categories:

1. SNXMAP Keywords
2. Common keywords for all Secure NetEx/IP applications
3. H214 GSK keywords
4. H624 and H804 OpenSSL keywords
5. H304 keywords

SNXMAP keywords	**Default**	**Definition**
BACKLOG	25	integer: number of outstanding connections (1-100)
DEBUG	Off	Boolean: enable/disable extra debug logging for SNXMAP
LOG	On	Boolean: enable logging to stdout/stderr
LOGFILE	Default	string: redirect stdout/stderr to named file (may be ignored on some platforms)
SYSLOG	Off	Boolean: enable syslog (where available)
SYSLOGFAC	local3	string: syslog facility name (where available)

Common keywords for all Secure NetEx/IP applications	Default	Definition
CCTABLE	none	string: path to code conversion table file
CMDFILE	none	string: path to command file
CNVERIFY	Off	Boolean: SSL certificate Common Name verification **UNIX:** Client will verify the remote server hostname matches SubjectName or SubjectAltName in server certificate OR will verify the remote IP address matches the SubjectAltName in the server certificate. **Unisys:** Ignore (N/A) **zOS:** Client will verify the remote server hostname matches the SubjectName in the server certificate.
CONNTO	30	integer: Connect timeout in seconds (0-300) (0=use system TCP connect timeout)
DEFBI	32768	integer: size of default input block (0-65535)
DEFBO	32768	integer: size of default output block (0-65535)
DNSRR	Off	Boolean: DNS round-robin (randomly select IP) This applies to all IP resolution methods (USEDNS, USESETIP).
IDLETO	6	integer: Utilizes TCP keepalive to implement idle time in seconds (0-disable) Valid range 1-7200 Unisys: Actual time is rounded to the up to the nearest 60 seconds.
LCLHOST	LCLHOST	string: local host name (max 8 characters) The local host name will be used internally in uppercase.
MAXBI	32768	integer: maximum size of input block (2048-size of integer-1)
MAXBO	32768	integer: maximum size of output block (2048-size of integer-1)

Common keywords for all Secure NetEx/IP applications	Default	Definition
MULTIHOST	Off	boolean: host name of connect and offer must match.
PORTNUM	none	integer: number of ports used for offers (1-65535) (PORTNUM+PORTSTART must be in the range 1-65535)
PORTSTART	none	integer: starting port used for offers (1-65535) (PORTNUM+PORTSTART must be in the range 1-65535)
RETRY-3501	On	boolean: try connect with next address on 3501 error. Enable if remote host name refers to a group of hosts. Disable if remote host name refers to a single host.
RETRY-3920	On	boolean: try connect with next address on 3920 error. Enable if local host has multiple addresses. Disable if local host has a single address. NOTE: 3920 is only possible if remote SNXMAP has USECONIP set.
SMWAIT	15	integer: wait time between offer lookup and connect (0-600)
USECONIP	Off	boolean: only incoming connections from a setconip address are allowed. SNXMAP will verify the source address when an offer is requested. On error, connect will fail with 3920 and offer remains active. Application will verify the source address during offer completion. On error, offer will fail with 3920. Connecting application will fail with 3201 or 3992.

Common keywords for all Secure NetEx/IP applications	Default	Definition
USEDNS	On	boolean: use system name services for host resolution If USEDNS and USESETIP are enabled SETIP data will be queried first. If SETIP data provides no hostname match then system name services are used. If SETIP does provide a hostname match, system name services will not be attempted.
USESETIP	Off	boolean: use data defined with SETIP commands for host resolution If USEDNS and USESETIP are enabled SETIP data will be queried first. If SETIP data provides no match then system name services are used. If SETIP does provide a hostname match, system name services will not be attempted.

H214 GSK keywords	Default	Definition
CIPHER	none	Sets the list of available ciphers (must use 4-character cipher numbers as documented by GSK)
FIPSLVL	1	Set SSL FIPS mode; 0-disabled, 1-3 is the gsk_fips_state.
KEYFN	none	string: certificate database file
KEYLBC	none	string: client certificate label
KEYLBS	none	string: server certificate label
KEYPW	none	string: certificate database password

H214 GSK keywords	Default	Definition
GSKPROTO	ALL	Set a list of SSL protocol versions to use when establishing a secure connection. Values are ALL, TLSV1, TLSV1.1, TLSV1.2. The availability of a specific protocol version is SSL library dependent. The actual protocol version used will be negotiated to the highest version mutually supported by the client and the server. The SSLv2 and SSLv3 protocols are deprecated and will never be used. Use ':' to separate values. A leading '+' means add protocol. A leading '-' means remove protocol. Set the value 'ALL' before using '-'. Example: Allow TLS 1.2 and higher. ALL:-TLSV1:-TLSV1.1

H624 and H804 OpenSSL keywords	Default	Definition
CAFILE	none	string: trusted CA certificate file containing PEM formatted certificates Used to verify certificates. Server will send client certificate request with CA names from this file.
CAPATH	none	string: trusted CA certificate directory containing PEM formatted certificate files using hash names Used to verify certificates.
CERTFILE	/usr/share/nesi/snetex/snx.crt	string: certificate file in PEM format
CIPHER	(empty: cipher list will use the OpenSSL library default list)	Sets the list of available ciphers for SSL sessions. string: cipher list string described in the OpenSSL documentation
FIPSMODE	Yes	Set SSL FIPS mode on(yes) or off(no).
KEYFILE	/usr/share/nesi/snetex/snx.key	string: certificate private key file in PEM format

H624 and H804 OpenSSL keywords	Default	Definition
SSLPROTO	ALL	Set a list of SSL protocol versions to use when establishing a secure connection. Values are ALL, TLSV1, TLSV1.1, TLSV1.2 and TLSV1.3. The availability of a specific protocol version is SSL library dependent. The actual protocol version used will be negotiated to the highest version mutually supported by the client and the server. The SSLv2 and SSLv3 protocols are deprecated and will never be used. Use ':' to separate values. A leading '+' means add protocol. A leading '-' means remove protocol. Set the value 'ALL' before using '-'. Example: Allow TLS 1.2 and higher. ALL:-TLSV1:-TLSV1.1

H304 keywords	Default	Definition
KEYIN	SNTX	This is the operator keyin. The valid commands are TERM, SWITCH and any SNXMAPOP command.
USE	OUTPUT	Use name of the breakpointed print file.

Notes:

- Some of the parameters documented above may not be included in the sample snxmap.cfg file ("***/usr/share/nesi/snetex/snxmap.cfg***"). It is the responsibility of the user to enter these values as necessary into the installation-specific copy of "snxmap.cfg" prior to starting Secure NetEx/IP.
- For Unix OS systems, after a re-install or upgrade install, it is possible to have a newer snxmap.cfg file. If so, it will be named snxmap.cfg.rpmnew. It is the responsibility of the user to merge or update snxmap.cfg to reflect any additions or deletions.

Appendix H: Secure NetEx/IP Command File

The Secure NetEx/IP command file is defined by the CMDFILE keyword in the Secure Netex/IP configuration file. The commands in the file are executed when SNXMAP starts.

Edit this file with the following recommendations:

1. One command per line.
2. Command format is the same as sxnmapop.
3. The characters '*', '#', '"', '!', '.'or '/' indicate comments.
4. All commands are case-insensitive.

```
# ************************************************************************
# Secure NetEx Application Mapper Commands
#
setip host1 10.1.2.1
setip host1 10.1.3.1
setip host2 10.1.3.2
#
setconip 10.1.3.1
```

Figure 22. Sample SNXMAP Command File

Notes:

- Not all snxmapop commands are supported in the command file.
- Commands must use full name (snxmapop minimal match is not used).

Appendix I: Secure NetEx/IP Tools

This section documents the Secure NetEx/IP tools shipped with the product.

If you have any questions on running these tools please contact support@netex.com

SNXMVGEN

This tool will generate data for testing purposes. It will prompt the user for parameters.

The prompt will look like this:

```
SNXMVGEN  V 3.0   12/10/12   65535 Max data
ENTER:
#SESS #BLOCKS SIZE  ODATA LOOPS DMODE VALIDATE HOSTNAME OFFRNAME
NNN   NNNNN   NNN   NNNN  HHHH  Y/N     HHHHHHHH OOOOOOOO
```

Sessions: The number of concurrent sessions to process. This must be equal to or less than the number of sessions used by SNXMVEAT. No default.

Blocks: The number of blocks of data to generate. No default.

Size: The size of the blocks of data to generate in bytes. No default

OData: The number of bytes of ODATA to generate. Typically, this parameter can be set to 0.

Loops: The number of times to send all of the blocks. No default.

Dmode: The DATAMODE to use when sending the blocks (Source, Destination). No default. ()

Value	Data type
0	Bit stream mode
1	Octet Mode
2	ASCII (8 bit)
3	EBCDIC
4	Reserved
5	BCD
6	Field-data (Unisys)
7	Display Data

Many datamodes are invalid because the Source and Destination data types are incompatible (i.e.0102).

Sample data modes of 0202 would be for an ASCII to ASCII character transfer.

0203 would be for an ASCII to EBCDIC character transfer.

0302 would be for an EBCDIC to ASCII character transfer.

0000 would be for a binary to binary transfer. The systems must have the same number of bits per byte for this test to work.

Validate: Should the content of each received block be validated. No default.

Hostname: The Secure NetEx/IP hostname to send to. No default.

OffrName: The SNXMVEAT Offer to connect to send the data. No default.

SNXMVEAT

This tool will read data generated by SNXMVGEN. It will prompt the user for parameters.

The prompt will look like this:

```
SNXMVEAT   V3.1   04/18/14   65535 max data
Enter:
#Sessions Validate OffrName HostName
NNN       Y/N      OOOOOOOO HHHHHHHH
```

Sessions: The number of concurrent sessions to process. This must be equal to or greater than the number of sessions used by SNXMVGEN.

Validate: Should the content of each block be validated. No default.

OffName: The OffrName SNXMVEAT will connect to for the test. No default.

Hostname: The Secure NetEx/IP hostname to receive from. No default.

Running SNXMVEAT and SNXMVGEN:

1. On the receiving side, execute SNXMVEAT (this MUST be started before SNXMVGEN).

   When you start the SNXMVEAT application, you will be prompted to specify the number of concurrent sessions and whether you want the application to validate those sessions. Enter the values separated by a space character, then hit ENTER.

   You will need to use CTRL-C (or let the offer(s) time out) to stop the SNXMVEAT application when your testing is completed.

   Example:

   ```
   shell_prompt# SNXMVEAT

   SNXMVEAT   V3.1   04/18/14   65535 max data
   Enter:
   #Sessions Validate OffrName HostName
   NNN       Y/N      OOOOOOOO HHHHHHHH
   1 n SNXMVEAT sunrise

   Making 1 offers of SNXMVEAT , validate 0, hostname SUNRISE
   ```

   OS2200 Example

   ```
   @xqt H304*release000.SNXMVEAT
   ```

   IBM Example

   See H214 installation section

2. On the sending host, execute SNXMVGEN.

The SNXMVGEN application will prompt you to enter a suite of values to use during the test. Enter the values separated by a space character, then hit ENTER.

For this example, we specified one (1) session of 99995 blocks of 32000 bytes with zero (0) ODATA, one (1) loop and specify zero (0) for the DMODE. There is no validation required. The following is an example of an execution of SNXMVGEN (user input is *italicized*).

```
shell_prompt# SNXMVGEN
SNXMVGEN  V 3.0   12/10/12   65535 Max data
ENTER:
#SESS #BLOCKS SIZE  ODATA LOOPS DMODE VALIDATE HOSTNAME OFFRNAME
NNN   NNNNN  NNN   NNNN  HHHH  Y/N      HHHHHHHH OOOOOOOO
1 99995 32000 0 1 0 n sunrise SNXMVEAT
```

OS2200 Example

```
@xqt H304*release000.SNXMVEAT
```

IBM Example

See H214 installation section

Once both processes are up and running, on the SNXMVGEN side, after specifying the desired parameters and hitting the <Enter> key, you will see:

```
1 ses, 99995 blocks, 32000 bytes/blk, 0 odata bytes,
 1 loops, datamode 0, validate 0, to SNXMVEAT  at SUNRISE
Connect: Status: 0,Ind: 0, Session: 1 Try: 1
```

On the SNXMVEAT side you should see output similar information to:

```
COffer: Status: 0,Ind: 1, Session: 1 Try: 0
```

When each loop completes, the SNXMVGEN side will output the stats for the finished loop:

```
Session 1:
325.5054 Mbits/s,  40.6882 Mbytes/s, 1333.3199 OPs/s, 75 Sec, 99999 Blks, 3199872256 Bytes
```

On the SNXMVEAT side, the output when the test completes is similar to:

```
CDisc: Status: 0,Ind: 0, Session: 1
Session 1:
325.5052 Mbits/s,  40.6881 Mbytes/s, 1333.3199 OPs/s, 75 Sec, 99999 Blks, 3199872256 Bytes
```

Appendix J: Unisys SSL TRACING

In the event of SSL connection problems, please gather the following information:

JOBLOG (If you are connecting to a remote system)

COMAPI PRINT FILE

- Before the job is submitted, enter using your comapi keyin:
  - <*keyin*> log high (Turns on logging)
  - <*keyin*> log close (Starts a new print cycle)
  - Run the job
  - <*keyin*> log close (Close the print cycle This is the file with the data)
  - <*keyin*> log off (Turns off logging)

CPCOM TRACE FILE

- Before the job is submitted, enter using your cpcomm keyin:
  - <*keyin*> trace api-ssl,medium (Trace these records)
  - <*keyin*> trace api-tcp,medium (Trace these records)
  - <*keyin*> trace network,medium (Trace these records)
  - <*keyin*> trace ssl,medium (Trace these records)
  - <*keyin*> trace ip,medium (Trace these records)
  - <*keyin*> trace tcp,medium (Trace these records)
  - <*keyin*> trace close (Starts a new trace cycle)
  - Run the job
  - <*keyin*> trace close (Close the trace cycle This is the file with the data)
  - <*keyin*> trace off (Turns off tracing)
- Print the trace file.
  - Execute SYSLIB*CPCOMM.LTA (To print the trace)
  - The trace file name was displayed at close time
    - I (analyze interactively)
    - !HEX (print data in HEX)
    - !STATUS (Do a STATUS)
    - !ALL (Print all trace Records)
    - !QUIT (End)
    - Send in the printed trace file. The name is displayed