

Network Control Program



X.25 Planning and Installation

NCP Version 7 and X.25 NPSI Version 3

Network Control Program



X.25 Planning and Installation

NCP Version 7 and X.25 NPSI Version 3

Note

Before using this document, read the general information under "Notices" on page ix.

Fourteenth Edition, July 2001

This is a revision of, and obsoletes, SC30-3470-12.

This edition applies to:

- Advanced Communications Function/Network Control Program (ACF/NCP) Version 7 Release 8.1 (program number 5648-063)
- Version 3 of the licensed program X.25 Network Control Program Packet Switching Interface (NPSI) (program number 5688-035)

and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. See "What Is New in This Book" on page xiii for the changes made to this document. Make sure you are using the correct edition for the levels of the products.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

Design and Information Development
Dept. CGF/Bldg. 656
International Business Machines Corporation
P.O. Box 12195
RESEARCH TRIANGLE PARK NC 27709-9990
USA

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information you supply in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1988, 2001. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	ix
Trademarks	x
About This Book	xi
X.25 Standard Definition	xi
Who Should Use This Book	xi
How to Use This Book	xi
What Is New in This Book	xiii
Where to Find More Information	xiii
NPSI Hardcopy Library	xiv
NPSI Softcopy Library	xiv
Chapter 1. X.25 Support Introduction	1-1
NCP X.25 ODLC Support for IBM 3746 Model 900	1-2
How to Install X.25 ODLC	1-3
Chapter 2. X.25 Support Installation and Generation	2-1
Relationship Between NCP and NPSI	2-2
What Is X.25 ODLC?	2-2
What Is NPSI?	2-2
How NPSI Fits into the Network	2-2
Overview of Network Definition Process	2-3
MVS Installation and Generation	2-4
VM Installation and Generation	2-8
VSE Installation and Generation	2-10
Chapter 3. Planning for an X.25 Network	3-1
Planning Concepts	3-3
Planning for the X.25 Interface	3-3
Planning for NPSI	3-18
Planning for NPSI and X.25 ODLC in the Network	3-45
Planning for NPSI and X.25 ODLC Migration	3-49
Planning for the Configurator	3-55
Chapter 4. Defining X.25 Networks	4-1
Physical Circuits	4-2
Logical Channels	4-2
Virtual Circuits	4-5
LLC Selection	4-23
CTCP Definitions	4-30
QOS (NPSI V3R4 and Later)	4-36
Fast Transaction Processing Interface (NPSI V3R4 and Later)	4-39
PAD Support	4-40
PCNE-to-PCNE Considerations	4-41
Multichannel Link Compatibility	4-42
Chapter 5. Resource Definition Reference	5-1
How Definition Statement and Keyword Information Is Presented	5-2
Resource Naming Conventions	5-2
Coding Conventions Used in This Chapter	5-6

BUILD Definition Statement	5-8
GENEND Definition Statement	5-12
SWMMT Definition Statement	5-13
X25.CPL Statement (NPSI V3R4 and Later)	5-14
X25.END Statement	5-21
X25.FCG Statement	5-22
X25.LCG Statement	5-30
X25.LINE Statement	5-31
X25.LU Statement	5-41
X25.MCH Statement	5-45
X25.NET Statement	5-80
X25.OUFT Statement	5-87
X25.PAD Statement	5-89
X25.PU Statement	5-90
X25.TRAN Statement	5-95
X25.VC Statement	5-97
X25.VCCPT Statement	5-111
Chapter 6. Coding NCP and VTAM Parameters for X.25 Support	6-1
NCP Considerations	6-2
VTAM Considerations for Switched Major Node	6-6
SVCSC Considerations	6-14
Session Continuation	6-18
Virtual Route Pacing for Subarea-to-Subarea Communication	6-19
Chapter 7. Controlling X.25 Resources	7-1
SNA Resources	7-2
Non-SNA Resources	7-5
Delayed Logical Link Activation (X.25 ODLC and NPSI V3R5 and Later)	7-11
Appendix A. Defining X.25 PSDN Facilities	A-1
Flow Control Parameter Negotiation	A-2
Called Line Address Modified Notification Facility	A-3
Priority of Data Facility (NPSI V3R4 and Later)	A-4
Called Address Extension Facility (AEF) (NPSI V3R4 and Later)	A-5
Closed User Group	A-5
Fast Select	A-6
High-Priority Class of Traffic	A-6
One-Way Logical Channel	A-7
Reverse Charging	A-7
Throughput Class Negotiation	A-8
Optional Facilities Using X.25 Statements	A-9
Optional Facilities Table	A-10
Appendix B. System Generation Examples	B-1
Example A—NPSI Basic Functions	B-2
Example B—NPSI Default and User-Defined IDNUM and Remote Loading	B-33
Example C—NPSI NPM and FTPI (V3R4)	B-96
Example D—NPSI HPTSS and Delayed Logical Link Activation	B-102
Example E—X.25/ISDN Interworking (NPSI V3R6 and Later)	B-117
Example F—Generation Information for X.25 ODLC	B-125
Appendix C. Statement and Keyword Changes	C-1
BUILD	C-2

X25.CPL (NPSI V3R4 and Later)	C-3
X25.END	C-4
X25.FCG	C-4
X25.LCG	C-5
X25.LINE	C-6
X25.LU	C-7
X25.MCH	C-8
X25.NET	C-11
X25.OUFT	C-12
X25.PAD	C-12
X25.PU	C-12
X25.TRAN	C-15
X25.VC	C-16
X25.VCCPT	C-18
Appendix D. Macros Shipped with NPSI	D-1
Appendix E. NPSI Error Messages	E-1
List of Abbreviations	X-1
Glossary	X-3
Bibliography	X-43
X.25 NCP Packet Switching Interface Publications	X-43
NCP, SSP, and EP Library	X-43
Other Networking System Product Libraries	X-44
Related Publications	X-45
Index	X-47

Figures

0-1.	Networking Symbols Used in This Book	xiii
1-1.	X.25 ODLIC Support	1-2
2-1.	The X.25 NDF Process	2-3
2-2.	X.25 Generation Summary under MVS	2-6
2-3.	Placement of X.25 Statements in NCP Generation (MVS)	2-7
2-4.	Placement of X.25 Statements in NCP Generation (VM)	2-10
2-5.	Placement of X.25 Statements in NCP Generation (VSE)	2-12
3-1.	Configuration Examples Supported by X.25 NPSI and TA	3-22
3-2.	ISDN Configuration Example with the IBM 3746 Model 900 and a Hitachi Terminal Adapter	3-25
3-3.	Logical Link Control Type 2	3-27
3-4.	Logical Link Control Type 3 (Peripheral Node)	3-29
3-5.	Logical Link Control Type 3 (Subarea Node)	3-30
3-6.	Logical Link Control Type 0	3-32
3-7.	Logical Link Control Type 4 (GATE)	3-33
3-8.	Fast Connect with One CTCP for One Physical Circuit	3-34
3-9.	Fast Connect with Two CTCPs for One Physical Circuit	3-35
3-10.	Association of SNA Session with X.25 Virtual Circuits for FTPI	3-36
3-11.	Logical Link Control Type 5 (Integrated PAD)	3-39
3-12.	Logical Link Control Type 5 (Transparent PAD)	3-40
3-13.	Virtual Circuits That Use DATE	3-41
4-1.	Logical Channel Definition	4-3
4-2.	Examples of Various IDBLK Values	4-13
4-3.	Fast Connect with Two CTCPs for One Physical Circuit	4-36
7-1.	Virtual Circuit and SNA Resource Assignment Order (Fast Connect Only)	7-8
7-2.	X.25 Virtual Circuit and SNA Resource Assignment	7-11
A-1.	NPSI Use of QOS Facility	A-4
X-1.	Example of a Language Statement	X-12
X-2.	Example of an NCP Definition Statement	X-12
X-3.	Example of a VTAM Definition Statement	X-12

Tables

2-1.	Library Names for NPSI	2-5
3-1.	Cause and Diagnostic Tables	3-10
3-2.	NCP Storage Requirements	3-19
3-3.	Recommended Limit for Number of Lines Sharing a CLP with X.25 Modulo 128 Lines	3-20
3-4.	Direction of a Virtual Call for the Dial Operation	3-22
3-5.	Functions Supported by NPSI on Interworking Connections	3-23
4-1.	IDNUM Assignment	4-12
4-2.	NPM Accounting Fields	4-20
4-3.	Number of Bytes Required for NPSI Performance Monitoring	4-21
4-4.	NPM Fields for MCH Link (Link Level)	4-21
4-5.	NPM Fields for MCH PU (Packet Level)	4-22
4-6.	NPM Fields for Virtual Circuit PU (Virtual Circuit Level)	4-22

4-7.	CUD0 Values and Associated LLC Types	4-24
4-8.	Use of Called Address Extension in Bits 0 and 1	4-26
4-9.	DIALNO or DLCADDR L Values	4-27
4-10.	LLC Numbers	4-28
4-11.	Results of Coding Example for CUD0 Value	4-28
4-12.	Enhanced Multichannel Link Compatibility	4-42
5-1.	NPSI Keywords for X25.CPL Statement	5-14
5-2.	VTAM Keywords for X25.CPL Statement	5-15
5-3.	NPSI Keywords for the X25.FCG Statement	5-22
5-4.	VTAM Keywords for the X25.FCG Statement	5-23
5-5.	X.25 Keywords for X25.LCG Statement	5-30
5-6.	X.25 Keywords for X25.LINE Statement	5-32
5-7.	VTAM Keywords for X25.LINE Statement	5-33
5-8.	X.25 Keywords for X25.LU Statement	5-42
5-9.	VTAM Keywords for X25.LU Statement	5-42
5-10.	X.25 Keywords for X25.MCH Statement	5-45
5-11.	VTAM Keywords for X25.MCH Statement	5-49
5-12.	Internal Clock Speed and LIC4 Configuration for Span of Available Line Speeds	5-73
5-13.	X.25 Keywords for X25.NET Statement	5-80
5-14.	X.25 Keywords for X25.OUFT Statement	5-87
5-15.	NPSI Keywords for X25.PAD Statement	5-89
5-16.	X.25 Keywords for X25.PU Statement	5-90
5-17.	VTAM Keywords for X25.PU Statement	5-91
5-18.	NPSI Keywords for X25.TRAN Statement	5-95
5-19.	X.25 Keywords for X25.VC Statement	5-97
5-20.	VTAM Keywords for X25.VC Statement	5-98
5-21.	X.25 Keywords for X25.VCCPT Statement	5-111
6-1.	PU Statement for a Physical Circuit	6-3
6-2.	PU Statement for a Permanent Virtual Circuit	6-4
6-3.	PU Statement for a Switched Virtual Circuit	6-4
6-4.	First LU Statement for a Physical Circuit	6-4
6-5.	Subsequent LU Statement for a Physical Circuit	6-5
6-6.	LU Statement for a Permanent Virtual Circuit	6-5
6-7.	DLCADDR Subfield ID Values	6-11
A-1.	Called Line Address Modified Notification	A-3
A-2.	Use of Called Address Extension in Bits 0 and 1	A-5
A-3.	NPSI Support	A-10
A-4.	Support for Implementation Requirements	A-11
C-1.	BUILD Changes	C-2
C-2.	X25.CPL Changes	C-3
C-3.	X25.END Changes	C-4
C-4.	X25.FCG Changes	C-4
C-5.	X25.LCG Changes	C-5
C-6.	X25.LINE Changes	C-6
C-7.	X25.LU Changes	C-7
C-8.	X25.MCH Changes	C-8
C-9.	X25.NET Changes	C-11
C-10.	X25.OUFT Changes	C-12
C-11.	X25.PAD Changes	C-12
C-12.	X25.PU Changes	C-12
C-13.	X25.TRAN Changes	C-15
C-14.	X25.VC Changes	C-16
C-15.	X25.VCCPT Changes	C-18

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Licensing
IBM Corporation
PO Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR

IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking	MVS/ESA
AIX	NetView
AnyNet	NTuneMON
APPN	NTuneNCP
BookManager	OS/390
eNetwork	OS/400
Enterprise System/9370	SecureWay
ESA/370	SystemView
ESA/390	System/370
ES/9370	Virtual Machine/Enterprise Systems
IBM	Architecture
IMS	VM/ESA
Library Reader	VTAM
LPDA	

TME 10 is a trademark of Tivoli Systems, Inc.

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

About This Book

NCP Version 7 and X.25 NPSI Version 3 Planning and Installation assists system programmers in planning and installing connections to X.25 packet switching networks from the IBM Network Control Program (NCP).

Two types of support are provided: one by the IBM Network Control Program (NCP) and the other by the IBM X.25 Network Control Program Packet Switching Interface (NPSI) licensed program.

- The Network Control Program (NCP) provides X.25 ODLC support when used in conjunction with the IBM 3746 Model 900 Connectivity Subsystem.

This support is provided starting with NCP V7R4 whether or not the IBM X.25 Network Control Program Packet Switching Interface (NPSI) licensed program is installed.

- The IBM X.25 Network Control Program Packet Switching Interface (NPSI) licensed program provides users with the ability to use communication facilities that support the X.25 Interface, as defined in “X.25 Standard Definition.”

NPSI V3R8 (and later) supports the IBM 3746 Model 900 Connectivity Subsystem.

NPSI support includes:

- Connections to SNA X.25 networks
- Connections to non-SNA X.25 networks

X.25 Standard Definition

The X.25 Interface is defined by the International Telegraph and Telephone Consultative Committee (CCITT), also known as the International Telecommunication Union — Telecommunication Standardization Sector (ITU-TS), at Geneva in 1976, Malaga-Torremolinos in 1984, Melbourne in 1988, Geneva in 1992 and Helsinki in 1993.

Who Should Use This Book

NCP Version 7 and X.25 NPSI Version 3 Planning and Installation is intended for system programmers who are responsible for planning for and installing X.25 support in a network.

You should be familiar with CCITT Recommendation X.25 or packet switching before reading this book, as well as with SNA concepts and products, as described in *Systems Network Architecture Concepts and Products*.

How to Use This Book

It is important that you read this book before you begin to plan for and install X.25 support in your network. This section includes information about the organization of the book and the symbols used in it.

Preface

This book contains an introduction as well as specific information on X.25 planning and installation. There is information on defining networks. There is a resource definition reference section that has information on X.25 definition statements and keywords. Information on coding NCP and VTAM parameters for X.25 is presented as well as information on controlling X.25 resources and defining X.25 PSDN facilities.

How CSS and 3746 Model 900 Are Used

The term *connectivity subsystem (CSS)* refers to the 3746 Model 900 Connectivity Subsystem, an expansion frame that extends the connectivity and enhances the performance of the IBM 3745 Communication Controller.

How Virtual Telecommunications Access Method (VTAM) Is Used

The term *VTAM* means either VTAM or OS/390 SNA Services, in other words, any of the following:

- VTAM for MVS/ESA, VM/ESA, or VSE/ESA
- The SNA Services feature of eNetwork Communications Server for OS/390
- The SNA Services feature of SecureWay Communications Server for OS/390

Symbols Used in This Book

Figure 0-1 on page xiii illustrates the networking symbols used throughout this book.

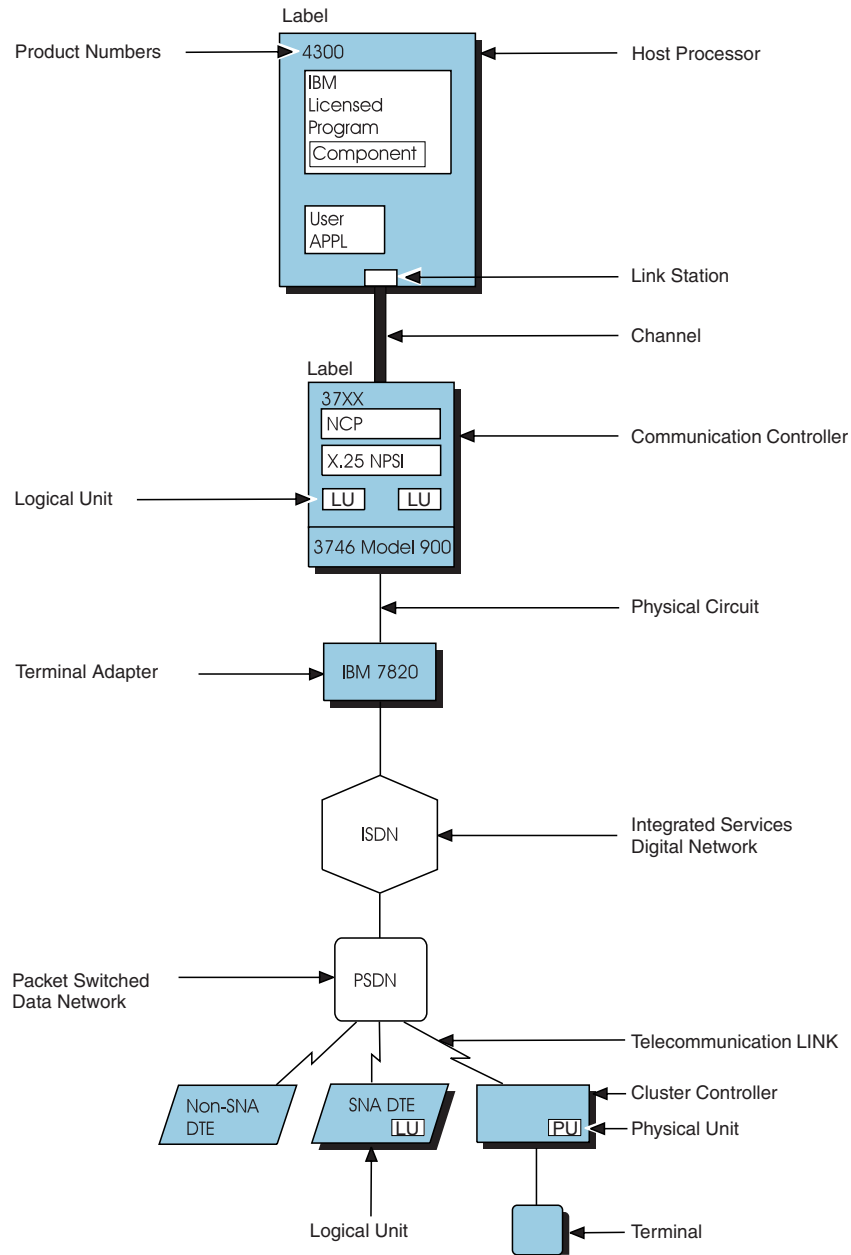


Figure 0-1. Networking Symbols Used in This Book

What Is New in This Book

New or changed technical material is marked by a vertical bar (|) in the left margin.

Where to Find More Information

For information about related publications, see the “Bibliography” on page X-43.

NPSI Hardcopy Library

The NPSI hardcopy library includes:

- *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning*

Note: This publication is also sent to NCP users who request X.25 support via a feature code.

- *X.25 NPSI Version 3 Release 9 Data Areas*
- *X.25 NPSI Version 3 General Information*
- *X.25 NPSI Version 3 Host Programming*

NPSI Softcopy Library

Those publications available as softcopy books have cross-document search and hypertext links for speedy, online information retrieval. These softcopy books are grouped together on an electronic bookshelf as part of the *ACF/NCP, ACF/SSP, EP, NPSI, and NTuneMON Softcopy Collection Kit*, LK2T-0414, on compact disc read-only memory (CD-ROM).

You can view and search these softcopy books by using BookManager READ products or by using the IBM Library Reader product included on the CD-ROM. For more information on CD-ROMs and softcopy books, see the *IBM Online Libraries: Softcopy Collection Kit User's Guide*, the BookManager READ documentation, or the BookManager home page at:

<http://booksrv2.raleigh.ibm.com>

You can also access unlicensed softcopy NCP publications in Acrobat or BookManager formats from the NCP home page at:

<http://www.networking.ibm.com/ncp>

Chapter 1. X.25 Support Introduction

NCP X.25 ODLC Support for IBM 3746 Model 900	1-2
Advantages to NCP Users	1-2
Advantages to NPSI Users	1-3
How to Install X.25 ODLC	1-3

This publication describes how to plan for and install NPSI and how to define X.25 resources.

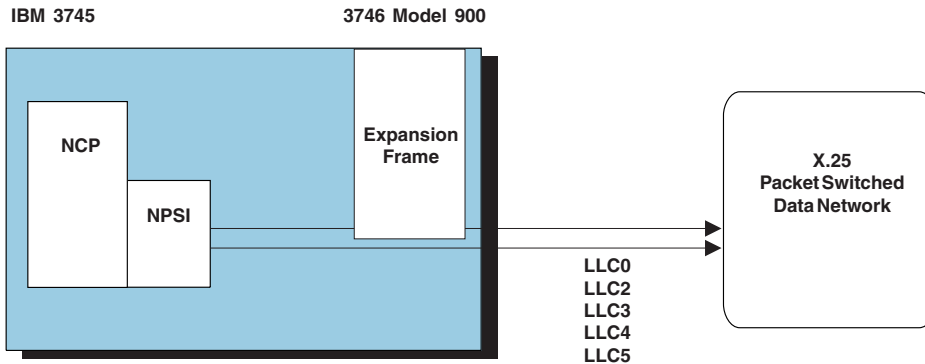
Prior to NCP V7R4, connections to X.25 networks were possible only through NPSI. Starting with V7R4, NCP has added support for connecting X.25 SNA resources using the outboard data link control (ODLC) of the IBM 3746 Model 900 Connectivity Subsystem. X.25 ODLC provides X.25 connection capabilities only for QLLC protocols (LLC3).

For more information about the IBM 3746 Model 900 connectivity subsystem, see the *NCP and EP Reference*.

NCP X.25 ODLC Support for IBM 3746 Model 900

Figure 1-1 shows that NCP provides support for X.25 ODLC devices when running with the IBM 3746 Model 900 Connectivity Subsystem. NCP handles PIU routing. The lower-level functions (QLLC, PLP, and LAPB) are provided by the CLP adapters in the 3746 Model 900 expansion frame. The NCP X.25 ODLC function supports only X.25 QLLC connections. NPSI is not required for X.25 ODLC connections.

NCP V7R3 and Earlier



NCP V7R4

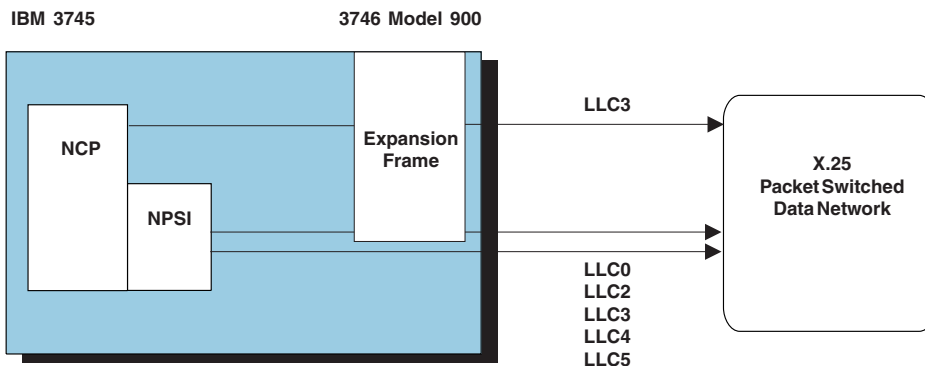


Figure 1-1. X.25 ODLC Support

Advantages to NCP Users

X.25 ODLC provides two advantages for NCP users of the IBM 3746 Model 900 systems:

- It provides support for X.25 SNA devices via QLLC protocols without requiring an NPSI license.
- It increases throughput by having the 3746 Model 900 expansion frame perform the lower-level processing.

Advantages to NPSI Users

Advantages to NPSI users are:

- Existing hardware and software environment is not affected by the new functions.
- IBM 3746 Model 900 users may choose to take advantage of X.25 ODLC to gain increased throughput.

How to Install X.25 ODLC

X.25 ODLC is a function of NCP V7R4 and later. To use X.25 ODLC you must perform network definition using the NCP/EP Definition Facility (NDF) of SSP V4R4 and later.

Refer to Chapter 2, “X.25 Support Installation and Generation” on page 2-1 for a description of the network definition process.

Refer to “Migration from NPSI V3R8 to X.25 ODLC” on page 3-54 for a description of the migration considerations.

Refer to Chapter 5, “Resource Definition Reference” on page 5-1 for a description of the keywords that apply to X.25 ODLC.

Example F in Appendix B shows sample NCP generation input for X.25 ODLC resources.

Chapter 2. X.25 Support Installation and Generation

Relationship Between NCP and NPSI	2-2
What Is X.25 ODLC?	2-2
What Is NPSI?	2-2
How NPSI Fits into the Network	2-2
Overview of Network Definition Process	2-3
MVS Installation and Generation	2-4
X.25 ODLC Installation	2-4
NPSI Installation	2-4
X.25 ODLC and NPSI Generation Process	2-5
Placement of X.25 Statements in NCP Generation	2-7
PTF Maintenance of X.25 ODLC and NPSI	2-7
VM Installation and Generation	2-8
X.25 ODLC Installation	2-8
NPSI Installation	2-8
X.25 ODLC and NPSI Generation Process	2-9
Placement of X.25 Statements in NCP Generation	2-10
VSE Installation and Generation	2-10
X.25 ODLC Installation	2-10
NPSI Installation	2-11
X.25 ODLC and NPSI Generation Process	2-11
Placement of X.25 Statements in NCP Generation	2-12

This chapter describes NPSI and explains the planning required for installing this IBM licensed program. For example, when planning to install NPSI, you must answer several questions, such as:

- What are the hardware and software requirements?
- What types of connections are required?
- How do you specify NCP statements?

Background information about the NPSI licensed program and packet switched data networks (PSDNs) can be found in *X.25 NPSI Version 3 General Information*.

This book assumes that you are familiar with the concepts and procedures needed to install IBM licensed programs, such as NCP. This knowledge will help you perform the necessary planning tasks for the NPSI installation.

Relationship Between NCP and NPSI

Starting with SSP V4R4 and NCP V7R4, the relationship between NCP and NPSI is changed in two ways:

- NCP provides X.25 ODLC support.
- The SSP installation tape includes X.25 NDF modules, and the NCP installation tape contains the X.25 NDF macros necessary for processing generation statements for X.25 ODLC resources. Therefore NDF statements for X.25 ODLC resources can be processed whether or not the NPSI licensed program is installed.

The following topics describe the relationship between NCP and NPSI, the X.25 ODLC support provided by SSP (starting with V4R4) and NCP (starting with V7R4), and how to install X.25 ODLC in an IBM 3746 Model 900 environment.

What Is X.25 ODLC?

X.25 ODLC is a function of NCP V7R4 and later that allows users to access X.25 packet switched data networks (PSDNs) within their existing network using the IBM 3746 Connectivity Subsystem. X.25 ODLC allows SNA host processors to communicate with SNA equipment over PSDNs that use X.25 protocols without using the X.25 NPSI licensed program.

What Is NPSI?

NPSI is an interface that allows SNA users to use X.25 PSDNs with their existing network. NPSI allows SNA host processors to communicate with both SNA and non-SNA equipment over PSDNs that use X.25 protocols.

How NPSI Fits into the Network

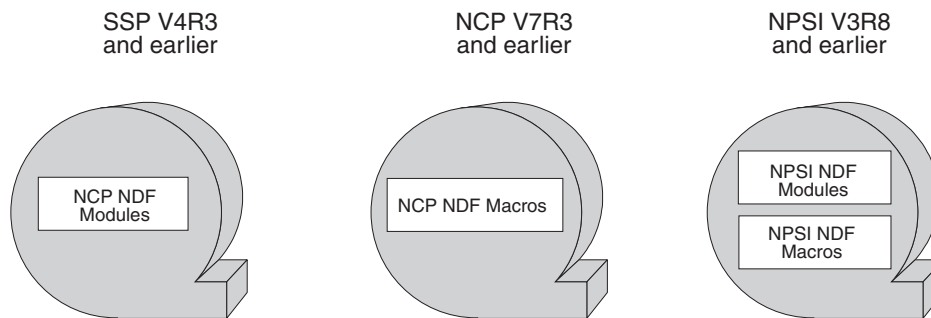
Within the communication controller, NPSI operates under the control of the NCP. The NCP/EP Definition Facility (NDF) processes the NCP and NPSI statements to create tables, VTAM source statements, and the link-edit control statements. The tables (in object form) and the linkage editor control statements, together with the preassembled modules supplied with NCP and NPSI program products, are input to the linkage editor. The linkage editor then creates a load module containing NCP and NPSI processes.

Overview of Network Definition Process

Figure 2-1 shows the NCP V7R4 and later generation process. NPSI NDF modules are now included on the SSP V4R4 and later installation tape. Certain X.25 NDF macros (those required for X.25 ODLC) are included on the NCP V7R4 installation tape. Therefore NDF statements for X.25 ODLC can be processed whether or not the NPSI licensed program is installed.

Starting with NPSI V3R9 the NPSI NDF modules and macros are no longer shipped with NPSI.

Previous NDF Process



New NDF Process

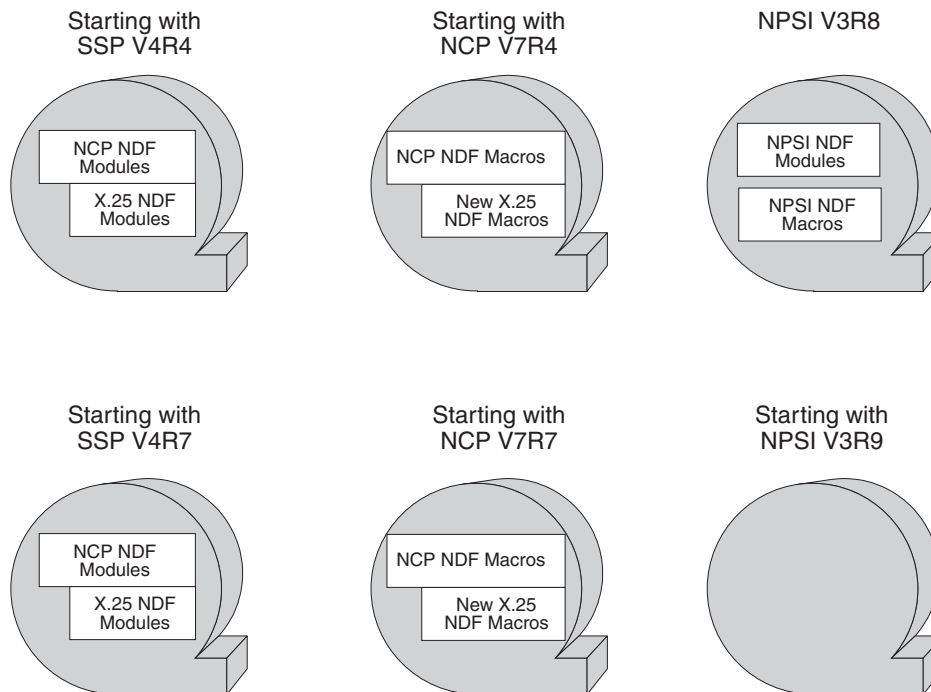


Figure 2-1. The X.25 NDF Process

MVS Installation and Generation

X.25 ODLC and NPSI generation under MVS occurs as a two-stage, inline process in the first step of NCP generation. X.25 ODLC and NPSI use the NDF standard attachment facility for generation.

X.25 ODLC Installation

X.25 ODLC resources can be generated and activated by installing NCP V7R4 and later, and SSP V4R4 and later. If you require only QLLC SNA line connections on the IBM 3746 Model 900, it is not necessary to install the X.25 NPSI licensed program.

For more information about the NCP installation process, see the *NCP Program Directory*.

NPSI Installation

NPSI is installed from tape. To select the correct tape files, see the instructions contained in the NPSI program directory. When installing NPSI under MVS, restore the following libraries from the NPSI distribution tape:

- SYS1.NPSIOBJ contains the object code for the NPSI part of the NDF. This library is used for maintenance for the NPSI part of the generation. This dataset is needed only when generating NPSI with NCP V7R3 or earlier (that is, when USERGEN=X25NPSI on the OPTIONS statement.)

Starting with NPSI V3R9, this library is not on the NPSI tape.

- SYS1.NPSILNK is created during the installation of NPSI and is used by NDF. This library is concatenated to the NCP SYS1.SSPLIB library.

Starting with NPSI V3R9, this library is no longer created.

- SYS1.SBALMAC1 contains the NPSI macros used during the combined NCP and NPSI generation. These macros are used during the table assembly of NCP generation.
- SYS1.SBALMOD1 contains the preassembled NPSI modules. This library is used during the link-edit step of the combined NCP and NPSI generation. You must code a DD statement with a DDNAME of ABALMOD1. The following statement specifies the system library of SBALMOD1 for the data set name to define the object module library at link-edit time:

```
//ABALMOD1 DD DSN=SYS1.SBALMOD1  
            DISP=SHR
```

See “NPSI Link Edit Step” on page B-12 for more information.

- ABALMOD1 is the distribution library for NPSI. This library contains all SMP ACCEPTed maintenance.

Table 2-1. Library Names for NPSI

Content	System Library	Distribution Library
Generation modules	none	NPSIOBJ ¹
Generation load module	NPSILNK ²	none
Generation macros	SBALMAC1 ³	ABALMAC1 ⁴
Executable modules	SBALMOD1 ³	ABALMOD1 ⁴

Note:

¹ Starting with NPSI V3R9, NPSIOBJ is not on the NPSI tape.

² NPSILNK is not delivered with MVS, but created during installation and used during NPSI generation. It is delivered with VM.

Starting with NPSI V3R9, NPSILNK is not created during NPSI generation, and is not on the VM tape.

NPSI has two sets of libraries:

³**S** For the system (target) library name.

⁴**A** For the distribution library name.

NPSI code is first installed in the system library, also known as the target library. When the code is successfully installed, it is copied into the distribution library at System Modification Program (SMP) ACCEPT time.

For more information about the NCP installation process, see the *NCP Program Directory*.

X.25 ODLC and NPSI Generation Process

The combined X.25 NCP generation procedure results in the following:

- One generation set contains both X.25 and NCP statements.
- Load module creation takes less time.
- The NCP and X.25 listing is printed as one item.
- You can use the NDF diagnostic aids.

To verify the input definitions without going through the entire generation process, use the FASTRUN keyword on the OPTIONS statement. When FASTRUN is selected, NDF reads the input and performs a validity check. Only the input section and error listing are generated. Control blocks and the load module are not produced.

Figure 2-2 on page 2-6 describes the production of the NCP/NPSI load modules. The NPSI statements described in Chapter 5, "Resource Definition Reference," provide the attachment to the PSDN. These statements are required for the NPSI part of the NCP generation and define the NPSI control blocks through the use of NDF.

The linkage editor uses both the NPSI library (SYS1.SBALMOD1) and the NCP library (SYS1.SNCPMOD1) as input. This produces the following:

- NCP/NPSI Load Module
- Resource Resolution Table

The NCP load module contains NCP and NPSI control blocks and executable code.

MVS Installation and Generation

TBL1OBJ and TBL2OBJ are small data sets. The size depends on the number of MCHs and virtual circuits. A safe order of magnitude is one cylinder of 3380 for every 200 virtual circuits.

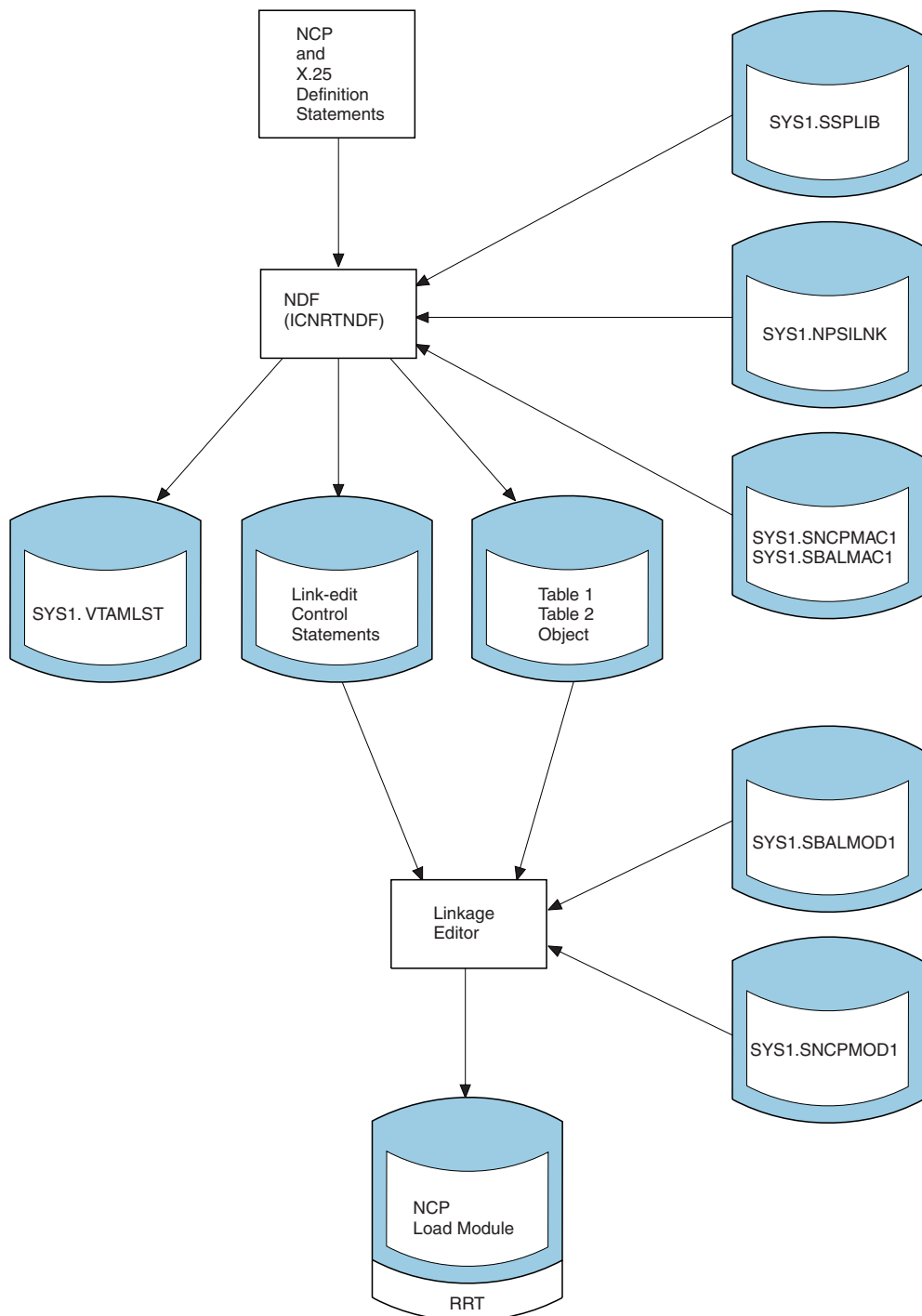


Figure 2-2. X.25 Generation Summary under MVS

Placement of X.25 Statements in NCP Generation

The X.25 generation process under MVS allows you to specify the X.25 definitions within the NCP definitions by using the NDF standard attachment facility. To accomplish this, code X.25 statements in the NCP generation code after all SS and BSC definitions, if any, and before all SNI NETWORK statements (nonnative network definitions). See Figure 2-3.

```

NCP BUILD Statement
.
.
Start Stop Definitions
.
.
BSC Definitions
.
.
Synchronous Data Link Control (SDLC) Definitions
.
.
X.25 Definitions
  X25.NET Statement
.
.
  X25.END Statement
XI Definitions (if XI is used)
.
.
Channel Adapter Definitions (required for 3745 with or without
the 3746 Model 900 attached)
.
.
SNI Definitions (nonnative network definitions)
.
.
GENEND Statement

```

Figure 2-3. Placement of X.25 Statements in NCP Generation (MVS)

PTF Maintenance of X.25 ODLIC and NPSI

System (target) libraries are defined for generation macros and executable modules. These libraries allow for the conditional installation of X.25 program temporary fix (PTF) maintenance. During the SMP APPLY step for PTFs, only the system (target) libraries are updated. To include the PTF maintenance that is in APPLY status in your 37XX load module, you must run the NDF generation process with the JCL pointing to the system (target) libraries, as shown in Appendix B, "System Generation Examples."

If you only want to include the PTF maintenance that is in ACCEPT status, you can modify your JCL to point to the distribution libraries and generate the 37XX load module. Using both the system and distribution libraries allows you to decide which of your 37XX load modules will include PTF maintenance in APPLY status.

If you decide that you no longer want the SMP maintenance that is in APPLY status, you have the option of doing an SMP RESTORE to remove this maintenance. SMP copies the previous or backup levels of the appropriate NPSI macros and modules from the distribution libraries to the system (target) libraries. You can rerun your NDF generation process to include the new maintenance level in your

VM Installation and Generation

37XX load module. After having run the PTF maintenance in APPLY status to ensure that it is error free, you can ACCEPT the SMP maintenance, which updates the NPSI distribution libraries, making the PTF maintenance permanent. After any SMP APPLY or RESTORE operation, always rerun your NDF generation steps to re-create your 37XX load module.

VM Installation and Generation

X.25 generation under VM occurs as a two-stage, inline process in the first step of the NCP generation.

X.25 ODLC Installation

X.25 ODLC resources can be generated and activated by installing NCP V7R4 and later, and SSP V4R4 and later. If you require only QLLC SNA line connections on the IBM 3746 Model 900, you do not need to install the X.25 NPSI licensed program.

For more information about the NCP installation process, see the *NCP Program Directory*.

NPSI Installation

NPSI is installed from tape. To select the correct tape files, see the instructions in the NPSI program directory. No specific NPSI EXECs are provided to perform NPSI generation. NPSI uses the NDF standard attachment facility for generation.

For a complete explanation of the NCP installation procedure, see the *NCP Program Directory*.

NPSILNK LOADLIB or CBEX25 LOADLIB, SBALMAC1 MACLIB, and SBALMOD1 TXTLIB must be defined in the VMNCP EXEC before you can run the NCP/NPSI generation.

For a complete explanation of the NCP generation and loading procedure, see the *NCP, SSP, and EP Generation and Loading Guide*.

NPSI Version 3 Release 8 and later are shipped using the VMSES/E packaging methodology. The VMSES/E component in VM/ESA provides a simplified installation and service methodology for VM software deliverables.

The instructions to install NPSI are included in the NPSI program directory. The NPSI product consists of the following libraries:

- For NPSI V3R8 and earlier, NPSILNK LOADLIB contains the load module used to generate NPSI.
Starting with NPSI V3R9, this library is no longer on the tape.
- SBALMAC1 MACLIB contains the NPSI macros used during the combined NCP and NPSI generation. These macros are used during the table assembly of NCP generation.
- SBALMOD1 TXTLIB contains the preassembled NPSI text decks. This file is used during the link-edit step of the combined NCP and NPSI generation. The DDNAME of the FILEDEF statement for the target library used to define the NPSI object modules library must be SBALMOD1.

X.25 ODLC and NPSI Generation Process

NCP V7R3 and Earlier: If you are implementing NPSI functions, you must define the following files in the VMNCP EXEC before you can run the NCP/NPSI generation:

```
NPSILINK LOADLIB
SBALMAC1 MACLIB
SBALMOD1 TXTLIB
```

NCP V7R4 and later: If you are implementing both NPSI and X.25 ODLC functions, you must define the following files in the VMNCP EXEC before you can run the NCP/NPSI generation:

```
CBEX25 LOADLIB
SBALMAC1 MACLIB
SBALMOD1 TXTLIB
```

NCP V7R4 and later: If you are implementing only the X.25 ODLC function, you must define the following file in the VMNCP EXEC before you can run the NCP/NPSI generation:

```
CBEX25 LOADLIB
```

For more information, see “Example for a NCP/PEP Generation with User-written Code using the NDF Attachment Facility” in the “Example of Execs for VM/SP” chapter of the *NCP, SSP, and EP Generation and Loading Guide*.

To verify the input definitions without going through the entire generation process, use the FASTRUN keyword on the OPTIONS statement. When FASTRUN is selected, NDF reads the input and performs the validity check. Only the input section and error listing are generated. Control blocks and the load module are not produced.

The TBL1OBJ and TBL2OBJ are small files. The size depends on the number of MCHs and virtual circuits. A safe order of magnitude is one cylinder of 3380 for every 200 virtual circuits.

The X.25 statements described in Chapter 5, “Resource Definition Reference,” provide the attachment to the PSDN. These statements are required for the X.25 part of the NCP generation and define the X.25 control blocks through the use of NDF.

For NPSI, the linkage editor uses both the NPSI object module library (SBALMOD1 TXTLIB) and the NCP object module library (SNCPMOD1 TXTLIB) as input. This produces the following:

- NCP/NPSI Load Module
- Resource Resolution Table

The NCP load module contains NCP and X.25 control blocks and executable code.

Placement of X.25 Statements in NCP Generation

The X.25 generation process under VM allows you to specify the X.25 definitions within the NCP definitions by using the NDF standard attachment facility. To accomplish this, code X.25 statements in the NCP generation code after all SS and BSC definitions and before all SNI NETWORK statements (nonnative network definitions). See Figure 2-4.

```
NCP BUILD Statement
.
.
Start Stop Definitions
.
.
BSC Definitions
.
.
SDLC Definitions
.
.
X.25 Definitions
  X25.NET Statement
.
.
  X25.END Statement
XI Definitions (if XI is used)
.
.
Channel Adapter Definitions (required for 3745 with or without
the 3746 Model 900 attached)
.
.
SNI Definitions (nonnative network definitions)
.
.
GENEND Statement
```

Figure 2-4. Placement of X.25 Statements in NCP Generation (VM)

VSE Installation and Generation

X.25 generation under VSE is an inline process that is part of NCP generation.

X.25 ODLC Installation

X.25 ODLC resources can be generated and activated by installing NCP V7R4 and later, and SSP V4R4 and later. If you require only QLLC SNA line connections on the IBM 3746 Model 900, you do not need to install the X.25 NPSI licensed program.

For a complete explanation of the NCP installation procedure, refer to the *NCP Program Directory*.

NPSI Installation

NPSI is installed from tape using the Maintain System History Program (MSHP). For more information, see the instructions in the NPSI program directory. The distribution tape contains the base product, as well as the desired usage tier. NPSI uses the NDF standard attachment facility for generation.

The installation of NPSI under VSE consists of the following:

- Restoring the production and generation sublibraries from the distribution tape
- Updating the VSE System History File

The following are the production sublibraries:

- NPSIV3Rx.PROD
- NPSIV3Rx.UTy

where:

- x Specifies 3, 4, 5, 6, 7, 8, or 9 depending on the release of NPSI being installed
- y Specifies the desired usage tier (2, 2.5, 3, 4, or 5)

For a complete explanation of the NCP installation procedure, refer to the *NCP Program Directory*.

X.25 ODLC and NPSI Generation Process

The X.25 ODLC function of NCP V7R4 and later can be implemented without the need to install NPSI. If X.25 NPSI functions are not required, refer to the *NCP Program Directory* for a complete description of the NCP installation procedure.

The combined NCP/X.25 generation procedure results in the following improvements:

- One generation set contains both NPSI and NCP statements.
- Load module creation takes less time.
- The NCP and X.25 listing is printed as one item.
- You can use NDF diagnostic aids.

To verify the input definitions without going through the entire generation process, use the FASTRUN keyword on the OPTIONS statement. When FASTRUN is selected, NDF reads the input and performs the validity check. Only the input section and error listing are generated. Control blocks and the load module are not produced.

The TBL1OBJ and TBL2OBJ are small files. The size depends on the number of MCHs and virtual circuits. A safe order of magnitude is one cylinder of 3380 for every 200 virtual circuits.

VSE Installation and Generation

The X.25 statements described in Chapter 5, “Resource Definition Reference,” provide the attachment to the PSDN. These statements are required for the NPSI part of the NCP generation and define the X.25 control blocks through the use of NDF. The linkage editor uses both the NPSI object module library and the NCP object module library as input. This produces the following:

- NCP/NPSI Load Module
- Resource Resolution Table

The NCP load module contains NCP and X.25 control blocks and executable code.

Placement of X.25 Statements in NCP Generation

The X.25 generation process under VSE allows you to specify the X.25 definitions within the NCP definitions by using the NDF standard attachment facility. To accomplish this, code X.25 statements in the NCP generation code after all start-stop (SS) and binary synchronous communication (BSC) definition if any, and before all SNA network interconnect (SNI) NETWORK statements (nonnative network definitions). See Figure 2-5.

```
NCP BUILD Statement
.
.
Start Stop Definitions
.
.
BSC Definitions
.
.
SDLC Definitions
.
.
NPSI Definitions
  X25.NET Statement
.
.
  X25.END Statement
Channel Adapter Definitions (required for 3745 with or without
the 3746 Model 900 attached)
.
.
SNI Definitions (nonnative network definitions)
.
.
GENEND Statement
```

Figure 2-5. Placement of X.25 Statements in NCP Generation (VSE)

Chapter 3. Planning for an X.25 Network

Planning Concepts	3-3
Planning for the X.25 Interface	3-3
CCITT Recommendation X.25	3-4
PSDNs	3-4
Planning PSDN Subscription	3-5
Packet Size	3-5
Packet-Level Window Size	3-6
Link-Level Window Size	3-6
Closed User Group	3-6
Reverse Charging	3-7
High-Priority Class of Traffic	3-7
Line Speed	3-8
PAD Service	3-8
Compliance with ISO Standards and CCITT 1984 (V3 Enhancements)	3-9
System Generation	3-9
Link Level	3-9
Packet Level	3-9
Compliance with ISO Standards and CCITT 1988 (V3R4 Enhancements)	3-14
System Generation	3-14
Link Level	3-14
Packet Level	3-15
Compliance with ISO Standards 7776 and 8208 (V3R5 Enhancements)	3-16
Link Level	3-16
Packet Level	3-16
Compliance with CCITT 1992 Standard	3-16
X.25 ODLC Compliance with ITU-T 1993 Standard	3-16
Planning for NPSI	3-18
Software and Hardware Compatibility	3-18
Storage Requirements	3-18
HPTSS (NPSI V3R5 and Later)	3-19
3746 Model 900 Connectivity Subsystem (NPSI V3R8 and Later)	3-19
Connection without a PSDN	3-20
X.25/ISDN Interworking Connection	3-21
Configuration	3-21
Establishing the Physical Path	3-21
Establishing LAPB and Virtual Circuits	3-21
Disconnecting the Physical Path	3-23
Functions Supported by NPSI on Interworking Connections	3-23
X.21 Leased Connections	3-23
X.21 Switched Connections (NPSI V3R6 and Later)	3-24
Logical Link Control (LLC) Types	3-26
SNA Connections	3-26
Non-SNA Connections	3-29
Virtual Circuits that Use DATE	3-38
Logical Link Control Type Selection	3-40
NPSI Support of RU Chaining for Long Non-SNA Messages	3-42
Inbound Flow	3-42
Outbound Flow	3-42
Quality of Service Facility	3-43
NPSI Support for QOS Facility	3-43

Priority Definition	3-43
Range Definition	3-43
Piggybacking Techniques	3-44
Link Level	3-44
Packet Level	3-44
Planning for NPSI and X.25 ODLC in the Network	3-45
SNA Considerations	3-45
PIU Segmentation	3-45
3270 Printer LU Type Selection	3-45
Pacing Parameters	3-45
Host Application Program Considerations	3-46
Non-SNA DTE Communication	3-46
Transparent PAD Implementation	3-46
CICS, IMS, and TSO Adjustments	3-46
Network Resource Recovery Considerations	3-48
NetView Performance Monitor	3-48
Accounting	3-48
Performance	3-49
Planning for NPSI and X.25 ODLC Migration	3-49
Migration from NPSI V1 to NPSI V3 Releases	3-49
Migration from NPSI V3R2 to NPSI V3R3	3-51
Migration from NPSI V3R3 to NPSI V3R4	3-51
Migration from NPSI V3R4 to NPSI V3R5	3-52
Calling DTE Address Verification	3-52
NCP V6R1 Currency	3-52
Migration from NPSI V3R5 to NPSI V3R6	3-53
X.25/ISDN Interworking	3-53
GATE MCH LU Processing	3-53
Migration from NPSI V3R6 to NPSI V3R7	3-53
Migration from NPSI V3R7 to NPSI V3R8	3-53
Migration from NPSI V3R8 to X.25 ODLC	3-54
Migration from NPSI V3R8 to V3R9	3-54
Planning for the Configurator	3-55

Planning Concepts

This chapter introduces you to the planning concepts needed for integrating X.25 into your network. The chapter is divided into five task sections:

- Planning for the X.25 interface
- Planning for NPSI
- Planning for NPSI and X.25 ODLC in the network
- Planning for NPSI and X.25 ODLC migration
- Planning for the configurator

Because NPSI and X.25 ODLC can connect SNA equipment through an X.25 PSDN, in addition to understanding SNA products, you should also understand Recommendation X.25, the PSDNs that conform to Recommendation X.25, and the subscription parameters for PSDNs that work with NPSI and X.25 ODLC.

As you plan your generation, you should also become familiar with the following:

- Hardware and software compatibility
- Logical link control types
- Storage requirements
- Connections that do not use PSDNs
- Conformance to ISO standards

Because NPSI and X.25 ODLC communicate with many components of the network, you need to understand:

- SNA considerations
- Host application considerations
- Network recovery considerations

If you used an earlier version of NPSI, you should also read “Planning for NPSI and X.25 ODLC Migration” on page 3-49.

Planning for the X.25 Interface

To plan for the use of X.25 in your network, you need to understand both Recommendation X.25 and the packet switched data networks (PSDNs) that are designed to conform to Recommendation X.25.

“CCITT Recommendation X.25” on page 3-4 gives you an introduction to Recommendation X.25.

“PSDNs” on page 3-4 gives you an overview of how the PSDNs work.

“Planning PSDN Subscription” on page 3-5 describes the specific subscription parameters available from PSDNs.

X.25 NPSI Version 3 General Information contains more information about X.25 and PSDNs.

CCITT Recommendation X.25

Recommendation X.25 was first approved by the CCITT in 1976. Since then, Recommendation X.25 has been refined and enhanced in 1980, 1984, 1988, 1992, and 1993.

Recommendation X.25 defines a standard interface for communication over three conceptual levels: physical, link, and packet.

- Physical level

The physical level specifies the electrical and physical characteristics of the interface (for example, voltage levels and circuit signals).

Recommendation X.21 from the CCITT defines the recommended physical interface. An alternative physical interface is defined in X.21 bis. X.21 bis is similar to the current V.24 (EIA RS-232-C in the U.S. and Canada) and V.35 interfaces implemented on IBM equipment.

- Link level

The link level converts the physical circuit into an error-free logical link used to transfer data between the data terminal equipment (DTE) and the data communication equipment (DCE).

Link level provides a standard protocol for the following procedures:

- Controlling the data link
- Checking the frame sequence
- Detecting the start and end of a frame
- Initializing the link
- Disconnecting the link
- Resetting the link

The address field defines the flow and direction of the frame. For example, the address field indicates whether the frame is a command or a response. Both the DTE and the DCE use the control field information to detect out-of-sequence frames, duplicated frames, missing frames, and the types of frames. The frame check sequence is a facility for detecting transmission errors.

- Packet level

The packet level defines the structure of *data* and *control* packets. Data packets contain the information that is to be transferred. Control packets are used to establish and manage virtual circuits.

The packet level specifies how a single physical circuit can be used by multiple logical circuits that handle simultaneous virtual calls. Packet flow over the logical circuit is also managed at the packet level.

PSDNs

PSDNs are networks connected by high-speed lines that carry data in small segments called *packets*. PSDNs perform the following functions:

- Route user-data packets
- Respond to command packets
- Monitor and manage the status of network resources

PSDNs and user-control data flow over physical circuits. Placing data into packets permits multiple connections to share one physical circuit. These connections,

which support many subscribers, can use the same PSDN simultaneously. Each connection is a logical point-to-point virtual circuit.

Each physical circuit can contain up to 4096 virtual circuits. Virtual circuits are created by connecting two end points of a network. Each end of the virtual circuit accesses the PSDN through a logical channel.

There are two types of virtual circuit connections supported in a PSDN:

- Permanent virtual circuit (PVC)
- Switched virtual circuit (SVC)

A PVC is analogous to a leased line in that a predefined association between two logical channels is made to the PSDN. An SVC is analogous to a switched circuit. A call setup procedure must be followed to establish a virtual circuit connection between two logical channels. The PSDN uses routing tables to maintain the relationships among the physical circuits, virtual circuits, and logical channels.

Note: Although access to the PSDN by a physically switched connection is part of the CCITT Recommendation X.25, most networks provide access only by physically nonswitched connections. NPSI and X.25 ODLC support only nonswitched connections to PSDNs.

Packets that flow on the virtual circuit have a maximum packet length. If the data to be sent has a length exceeding the maximum packet length, the data must be split into packets that do not exceed the maximum packet length. The PSDN then transmits the packets to the specified destination and guarantees delivery in the proper order with no information loss.

To ensure the proper delivery of packets, the PSDN uses a *packet header* that includes the logical channel identifier, a sequence number, and other control information. In most cases, the PSDN receives the packetized data from a DTE, sends an acknowledgment to the origin, and passes the packets to their destination.

Note: CCITT Recommendation X.25 applies to the interface between the PSDN DCE and a user DTE. PSDNs that provide an X.25 interface do not always operate in exactly the same way. In addition, not all PSDNs provide an X.25 interface.

Planning PSDN Subscription

The following considerations should be discussed with your PSDN vendor when you plan the attachment requirements.

Packet Size

Packet size is defined to the PSDN as the maximum size packet that can be transmitted between the DTE and DCE. NPSI and X.25 ODLC support packet sizes ranging from 32 through 4096 bytes, but many PSDNs have their own size restrictions. To ensure compatibility, discuss packet size with your network vendor. A larger packet size decreases central control unit (CCU) utilization by NPSI because splitting the path information unit (PIU) into several packets is time-consuming.

Packet size can be predetermined or negotiated at call setup.

Packet-Level Window Size

Packet-level window size defines the maximum number of packets that can be transmitted between the DTE and DCE on a particular virtual circuit, before an acknowledgment must be received. A packet window exists for each transmission direction. The packet-level window size affects end-to-end flow control for a virtual circuit and depends upon the following:

- Type of application
- Traffic flow
- Line access speed

Many PSDN vendors do not offer packet-level window size as a subscription parameter. Instead, they set the packet-level window size default at 2, because, as the packet-level window size increases, the buffer requirements within a PSDN increase.

Subscribing to or negotiating a packet window size of more than 2 (if the size is available from the PSDN vendor) increases the degree of piggybacking possible at the X.25 packet level, and therefore decreases CCU utilization.

Link-Level Window Size

Link-level window size defines the number of packets that can be transmitted on a physical circuit before an acknowledgment must be returned. This window size is the sum of all packets transmitted for all virtual circuits operating on a physical circuit. Link-level window size is a mechanism used to regulate information flow across the X.25 DTE/DCE interface.

For access lines with smaller error rates, you can obtain better throughput by defining larger link-level window sizes. For most PSDNs, the link-level window size is fixed at 7.

Closed User Group

A closed user group (CUG) is a PSDN facility that prevents unauthorized access to a DTE. When CUG is in use, switched logical channels not subscribing to the same CUG are unable to communicate with each other. The CUG option does not affect NPSI performance.

CCITT defines the following types of CUGs within X.25:

- Outgoing access
- Incoming access
- Incoming calls barred
- Outgoing calls barred
- Outgoing access selection
- Bilateral closed user group selection

Some networks allow all types of CUGs; others allow only some types. Check with your PSDN vendor to determine the CUGs that are available. NPSI and X.25 ODLC *unconditionally* accept CUGs for incoming calls by default. If you want to control CUGs, you must use the NPSI GATE or DATE function, or specify the type of CUG to be rejected in the RFAC keyword of the X25.NET statement. See *X.25 NPSI Version 3 Host Programming* to learn more about programming GATE and DATE.

Reverse Charging

Reverse charging for an X.25 connection is the same as reverse charging in the telephone network. With reverse charging, the called party pays the connection cost. PSDN vendors charge for connection in different ways. Charges can be determined by the number of virtual circuits and by the number of physical circuits. Many vendors allow you to specify reverse charging only by physical circuit, which will affect all virtual circuits using that physical circuit.

Some PSDN vendors determine charges using a volume-related charge that applies to a subscription account, rather than a virtual circuit. Each installation must pay a subscription charge. In this case, you can avoid multiple subscription charges by using reverse charging for one installation with a subscription account and have all packets charged to that single subscription.

Some PSDN vendors automatically charge the called party when a PSDN packet assembler/disassembler (PAD) is accessed.

You must arrange with the PSDN vendor at the time of subscription to either accept or reject reverse charge calls.

In some cases, PSDNs behave differently depending on whether you subscribe to reverse charging. Some PSDNs provide a Facility field when the Facility field is not present in the Call Request. In this case, the Facility field parameter is set to *reverse charging not requested*. Other PSDNs include the Facility field only when the Call Request specifies *reverse charging requested*. You should check with your PSDN vendor to determine how the PSDN handles reverse charging.

NPSI and X.25 ODLC *unconditionally* accept reversed charging of incoming calls by default. To control reverse charging, you must use the NPSI GATE or DATE function or specify that you want to reject reversed charging in the RFAC keyword of the X25.NET statement. See *X.25 NPSI Version 3 Host Programming* to learn more about programming GATE and DATE.

High-Priority Class of Traffic

This high-priority class of traffic service ensures that packets marked as high-priority are transmitted before other packets. However, this service requires the use of a smaller packet size and is done at a premium cost.

NPSI and X.25 ODLC fully support high-priority class of traffic service. NPSI checks incoming calls for high-priority service requests and internally updates the packet length to 128 for the duration of a high-priority call. For outgoing calls, you can specify priority service using the optional facilities keyword (OPTFACL) of the X25.OUFT statement. You must also code an associated X25.VCCPT statement defining a maximum packet length of 128.

See Chapter 5, “Resource Definition Reference,” for more information about the X25.OUFT and X25.VCCPT statements.

Note: Do not confuse high-priority class of traffic with the CCITT fast select facility.

Line Speed

Line speed is the number of bits per second (bps) that a physical circuit can transmit data. Greater line speed means an increased connection cost, in addition to the increased circuit cost. Lower line speed lengthens response time. To shorten response time, the average load on a physical circuit's transmit leg or receive leg should not exceed 60 percent. Plan your line speed needs carefully, and then discuss them with your PSDN vendor.

Note: The configurators for the 3720 and the 3745 (with or without the 3746 Model 900 attached) compute the use of each MCH leg (transmitting and receiving) as a function of the traffic described on each virtual circuit belonging to this MCH. This information can help in determining the required line speed.

V3R4 and Later: NPSI supports a 256 Kbps transmission line speed. You can use this support by assigning the appropriate values to the SPEED keyword on the X25.MCH statement. For more information on the X25.MCH SPEED keyword, see "X25.MCH Statement" on page 5-45.

V3R5 and Later: NPSI V3R5 supports a line speed up to 2048 Kbps for the HPTSS duplex SDLC adapter. This is done by assigning the appropriate values to the SPEED keyword on the X25.MCH statement. For more information about the X25.MCH SPEED keyword, see "X25.MCH Statement" on page 5-45.

X.25 ODLC and NPSI V3R8 and later: Line speeds up to 2048 Kbps are supported for the 3746 Model 900 with LIC 12.

PAD Service

A PSDN can include PAD service in its configuration. PAD service provides protocol conversion for connecting non-X.25 devices to an X.25 interface. CCITT has defined the following three PAD recommendations:

- Recommendation X.3 defines a set of parameters that can be used to select the operational characteristics of the PAD service.
- Recommendation X.28 defines the procedures that are used by the start-stop ASCII device that accesses the PAD. These procedures provide you with an initial set of X.3 PAD parameters, a profile, and an explanation of how these PAD parameters can be changed before or during each virtual call.
- Recommendation X.29 defines the procedures that access the PAD service, which can be used by the DTE. Recommendation X.29 also includes procedures to change the X.3 PAD parameters during the virtual call, if required.

NPSI provides two types of PAD support. The first type is called integrated PAD support. Integrated PAD support employs a subset of Recommendation X.29. An ASCII terminal conforming to X.28 and communicating through an X.3 PAD can use integrated PAD support to access an SNA host. The second type of PAD support is called transparent PAD support. With this type of support, the host application is responsible for those commands that allow communication with the remote PAD. Transparent PAD support is required for any PAD that does not follow Recommendations X.3, X.28, and X.29. Transparent PAD support can also be used with a PAD that supports the CCITT Recommendations X.3, X.28, and X.29 if the host application requires control of the PAD service in a different way than the support provided by integrated PAD support.

See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for a detailed explanation about how to use and customize PAD parameters.

Compliance with ISO Standards and CCITT 1984 (V3 Enhancements)

This section describes how NPSI complies with International Organization for Standardization (ISO) standards 7776 and 8208.

System Generation

Keywords must be added to the X25.NET statement to support the optional features at the packet level. See “Packet Level” for the keywords.

Keywords must be added to the X25.MCH statement to support the optional features at the link level. See “Link Level” for the keywords.

Link Level

The link level changes bring NPSI in closer compliance with optional features in the ISO 7776 and CCITT 1984 Recommendations. The changes include:

- After receiving a frame that is not valid, NPSI transmits an FRMR frame after the T1 timer elapses. NPSI will continue to send FRMR frames for the number of times specified by the NPRETRY keyword on the X25.MCH statement (CCITT N2 retry counter).
- NPSI immediately transmits an SABM, if an FRMR frame is received indicating frame rejection.
- NPSI transmits an FRMR frame and starts the T1 timer when an I or S frame that is not valid is received. If NPSI receives an FRMR frame before the T1 timer elapses, NPSI stops the T1 timer and immediately sends an SABM.
- If an MCH has been inactive for a longer time period than defined by the T4 timer, NPSI sends a Receive Ready (RR) frame, with the polling bit (P-bit) on. NPSI resets the T4 timer and continues to transmit a Receive Ready frame every time the timer elapses.

The T4 timer is defined in seconds during NPSI generation.

Packet Level

Enhancements that allow NPSI to comply more closely with ISO 8208 and CCITT 1984 at the packet level include:

- Use of CCITT 1984 diagnostic codes
- NPSI facility checking for Incoming Call and Call Connected packets
- Nonstandard NPSI facilities process (for Incoming Call and Call Connected packets)
- Additional validity checking of received packets
- Retries on Clear, Reset, and Restart packets
- Delivery Confirmation (D-Bit) acknowledgment
- Acceptance of duplicate facilities
- Logical channel number attribution
- Handling of call collision

- Consideration of non-full data packets
- Consideration of Qualified-bit (Q-bit) in complete packet sequences
- Interrupt packet timeout
- Facility checking for Call Accepted and Call Connected packets
- Allowing for different packet and window sizes
- Reset packet handling

Notes:

1. NPSI does not perform facility checking on Clear Indication packets or Clear Confirmation packets.
2. Modifications in the packet level section do not apply when using the DATE and GATE functions.

Use of CCITT 1984 Diagnostic Codes: When NPSI sends a Restart, Clear, or Reset packet to the network, it appends cause and diagnostic codes to the created packet. NPSI selects the appropriate cause and diagnostic codes to send to the DCE, depending on the specification of the CAUSE keyword on the X25.NET statement. See "X25.NET Statement" on page 5-80 for a description of the CAUSE keyword.

During NPSI system generation, a cause and diagnostic table is built according to how the CAUSE keyword is coded on the X25.NET statement. Table 3-1 shows the tables that can be built.

Table 3-1. Cause and Diagnostic Tables

Table	Module Name	Description
CCITT	BALCCITT	Cause 00 with CCITT 1984 diagnostics
IBM00	BALIBM00	Cause 00 with SNA diagnostics (for IBM migration)
IBM80	BALIBM80	Cause 80 with SNA diagnostics (same as IBM00)
DCECD	BALDCECD	Nonzero cause with same diagnostics as CCITT

The default value is:

CAUSE=IBM00

See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information about CCITT cause and diagnostic codes.

Note: Many IBM diagnostics do not have an equivalent CCITT diagnostic code. In such cases, even with CAUSE=CCITT specified, the IBM00 set of codes is sent.

NPSI Facility Checking for Incoming Call and Call Connected Packets: NPSI checks the format of the facility field in both the Incoming Call and the Call Connected packets.

NPSI accepts the following standard facilities without special coding:

Code	Facility
X'01'	High-priority class of traffic, fast select, or reverse charging
X'02'	Throughput class negotiation

X'03'	Closed user group—basic
X'08'	Called line address modified (V3R4 and later)
X'09'	Closed user group with outgoing access—basic
X'41'	Bilateral closed user group selection
X'42'	Packet size negotiation
X'43'	Window size negotiation
X'47'	Closed user group—extended
X'48'	Closed user group with outgoing access—extended
X'C3'	Call redirection notification
X'C9'	Called address extension facility (V3R4 and later)
X'CB'	Calling address extension facility
X'D2'	Quality of service negotiation (priority) (V3R4 and later)

Note: Facilities other than those listed previously are nonstandard and must be specified on the NSTDFAC keyword.

If the facility was not specified on the NSTDFAC keyword, or if these facility codes were not specified, NPSI clears the call. A Clear Request packet is sent if a format error is found.

NPSI checks the facility field format for a:

- Facility length greater than the remainder of the packet
- Facility length greater than 109 bytes
- Duplicated facility

A table of up to twelve predefined CCITT facilities can be built at system generation, and you decide the facilities, if any, that you want to reject. Use the keyword RFAC in the X25.NET statement to code the facilities you want to reject. For example:

```
RFAC=(fac1,fac2,fac3,...)
```

where fac is the mnemonic code of the facility to reject. Twelve facilities are available. Each facility is optional and can be rejected. The default value is NONE, indicating that no facility is rejected. See “NPSI Facility Checking for Incoming Call and Call Connected Packets” on page 3-10 for the list of facilities that you can reject.

Note: Fast select is supported only when communicating through a GATE CTCP.

When an Incoming Call or Call Connected packet containing optional facilities is received, and one of the facilities is coded for rejection, a Clear packet is sent to the network.

Nonstandard NPSI Facilities Process: In addition to the standard CCITT X.25 user facilities, you can generate a table of nonstandard facilities to be accepted. Although the nonstandard user facilities are not defined in the CCITT Recommendation X.25, they must have the proper facility format.

Use the NSTDFAC keyword in the X25.NET statement to code the nonstandard facilities. The default value is NONE. For example:

```
NSTDFAC=(Nsfac1,Nsfac2,Nsfac3...)
```

A nonstandard facility can be duplicated in a Call packet. In this case, the facility must be coded *as many times* as it might be duplicated. For example:

```
NSTDFAC=(Nsfac1,Nsfac2,Nsfac2,Nsfac2,Nsfac3...)
```

Additional Validity Checking of Received Packets: When a packet is received, NPSI determines whether the:

- Packet is valid in the current virtual circuit state
- Length of the packet is valid
- Packet has a valid format
- Virtual circuit number allows for the received packet type
- Packet is valid on the virtual circuit number

Retries on Clear, Reset, and Restart Packets: Timer values and retry counts are specified in the X25.NET statement. The default value for the timers is set at 180 seconds, with the exception of T21, which has a default of 200 seconds, and T25, which has a default of 0 seconds. The retry count default value is set at 1 for all retry counters, except for R21 (Call) where no retry is attempted.

The system generation keywords specified in the X.25 NET statement are:

Timer Values

T20	Restart request response timer
T21	Call request response timer
T22	Reset request response timer
T23	Clear request response timer
T25	Window rotation timer
T26	Interrupt confirmation timer

Retry Counters

R20	Restart request retries
R22	Reset request retries
R23	Clear request retries

Delivery Confirmation Bit (D-Bit) Acknowledgment: If you code DBIT=YES on the X25.MCH statement, NPSI acknowledges the presence of the D-bit for the PCNE function (LLC type 0) only.

For an incoming call, upon receipt of an Incoming Call packet with the D-bit set on, NPSI transmits a Call Accepted packet with the D-bit set on. For an outgoing call, NPSI transmits a Call Request packet with the D-bit set on, and waits for a Call Connected packet with the D-bit set on. After the Call Connected packet is received, the data packets from both directions have the D-bit set on. If NPSI transmits a Call Request packet having the D-bit set on, but receives a Call Connected packet with the D-bit set off, the data packets should not have the D-bit set on. However, if the data packets have the D-bit set on, NPSI does not reset the logical channel, but accepts the data packets and ignores the D-bit.

If NPSI receives a data packet with the D-bit set on over a GATE virtual circuit (LLC type 4), a RESET command is sent to the CTCP with a diagnostic code of X'A6', and the virtual circuit is reset. The CTCP is responsible for ignoring the error, sending a RESET command to the network, or clearing the connection.

Acceptance of Duplicate Facilities: If you code NETTYPE=4 on the X25.NET statement during system generation, NPSI rejects duplicate facilities and clears the call. However, if you code NETTYPE=1 or NETTYPE=3, NPSI accepts the duplicate facility. When a facility code appears more than once, NPSI uses the last one found.

Note: When DATE or GATE is used on the MCH, NPSI accepts all functions without regard to NETTYPE, and processes the Incoming Call packet to the CTCP. It is the responsibility of the CTCP to process the call properly.

Logical Channel Number Attribution: When NPSI originates a virtual call, it must choose a logical channel. To avoid call collision, NPSI DTE chooses a channel starting at the high end of the range of the logical channels.

Handling of Call Collision: Call collision occurs when a DTE transmits a Call Request packet over a logical channel and receives an Incoming Call packet over the same logical channel.

To avoid call collision, NPSI performs one of the following functions:

- When NPSI acts as a DTE, it holds and does not recognize the Incoming Call packet until it receives acknowledgment from the DCE.
- When NPSI acts as a DCE, it cancels the call request by sending an INOP link message to VTAM for a non-GATE call. For a GATE call, NPSI sends a CLEAR command to the CTCP in response to the Call command. The Incoming Call packet is processed.

Consideration of Non-Full Data Packets: NPSI considers a non-full data packet not valid if the more data bit (M-bit) is set on and the D-bit is off. If NPSI acts as a DTE, the logical channel is reset with a diagnostic code of X'A5'. When NPSI acts as a DCE, the M-bit is reset. The reset X'A5' can be avoided by coding the MBNOFULL=YES keyword on the X25.MCH statement.

Consideration of Qualified Bit (Q-bit) in Complete Packet Sequences: When NPSI receives a complete packet sequence with a qualified bit (Q-bit) discrepancy, the logical channel is reset with a diagnostic code of X'5E'.

Interrupt Packet Timeout: NPSI can send an Interrupt Request packet for GATE, DATE, integrated PAD, or transparent PAD. To receive interrupt confirmation for DATE and integrated PAD, you must start the T26 timer. However, if the timer expires for DATE and integrated PAD, NPSI transmits a Reset packet containing a diagnostic code of X'91'.

Facility Checking for Call Accepted and Call Connected Packets: NPSI sets the address and facility length fields to zero in the last two bytes of the Call Accepted packet. NPSI accepts a Call Connected packet even if the packet is not formatted with the address and facility length fields in the last two bytes.

Allowing for Different Packet and Window Sizes: Nonstandard packet and window sizes are optional user facilities that are defined using the NSTDFAC keyword on the X25.NET statement during system generation. You can use this facility to select packet and window sizes for each direction of data transmission from a list of default packet and window sizes.

NPSI provides an option to allow negotiation, for individual calls, of the default packet or window sizes through flow control negotiation.

This facility allows NPSI to propose window and packet sizes through flow control parameters, as well as the ability to adjust packet and window sizes in accordance with the parameters that are returned from the DCE.

Reset Packet Handling: The following information should be considered when a Reset packet is received by NPSI.

- For a PVC:
 - When a Reset packet is received, if RESETINO=YES is coded on the X25.NET statement, NPSI confirms the reset and sends an INOP message for the virtual circuit PU. However, if RESETINO=(ccdd,ccdd,...) is coded, the INOP message is sent for the virtual circuit PU only if the reset cause and diagnostic codes are not found in the ccdd list. If RESETINO=NO is coded, the INOP message is not sent for the virtual circuit PU.
 - When a Reset packet is sent, NPSI waits for a reset confirmation and sends an INOP message for the virtual circuit PU if RESETINO=YES or RESETINO=(ccdd,ccdd,...) is coded, *and* if reset cause and diagnostic codes are not found in the ccdd list. If RESETINO=NO is coded, the INOP message is not sent for the virtual circuit PU.
- For an SVC:
 - When a Reset packet is received and RESETINO=YES is coded on the X25.NET statement, NPSI confirms the reset and clears the virtual circuit. When NPSI receives a Clear Confirmation packet, it deactivates the virtual circuit PU. If RESETINO=(ccdd,ccdd,...) is coded, NPSI clears the virtual circuit PU only if the cause and diagnostic codes of the Reset are not found in the ccdd list. If RESETINO=NO is coded, the virtual circuit is not cleared.
 - When a Reset packet is sent, NPSI waits for a reset confirmation. If RESETINO=YES or RESETINO=(ccdd,ccdd,...) is coded, and if the cause and diagnostic codes of reset are not found in the ccdd list, NPSI sends a Clear packet. When NPSI receives confirmation, it causes the virtual circuit PU to become inoperable. If RESETINO=NO is coded, NPSI does not clear the virtual circuit PU.

In all cases (PVC and SVC), if RESETINO=YES or RESETINO=(ccdd,...), the INOP is done unless NPSI is processing a packet sequence with the M-bit on for LLC type 2 and LLC type 3.

Compliance with ISO Standards and CCITT 1988 (V3R4 Enhancements)

This section describes how NPSI V3R4 complies with ISO standards 7776 and 8208.

System Generation

Keywords must be added to the X25.NET statement to support the optional features at the packet level. T3 and T4 keywords have been added to the X25.MCH statement, and the default for the DM keyword has changed. See “Link Level” for the keywords.

Link Level

The link level changes bring NPSI in closer compliance with ISO 7776 and CCITT 1988 recommendations. The changes include the following:

- Since the 1984 CCITT conference, all networks using LAPB must support the disconnect mode (DM) response. For V3R4 and later, the DM keyword on the

X25.NET statement becomes optional, with a default of YES (DM supported by network).

- When NPSI is defined as a DCE, it supports an optional T3 timer system parameter, with the value of the T3 timer passed to the DTE. The T3 timer should be sufficiently greater than the T1 timer to ensure that when the T3 timer expires, the data link channel is in a nonactive, nonoperational state. This is necessary because the normal data link operation cannot resume when the data link channel requires a data link setup.
 - The T3 timer value must be greater than the T4 timer value, because the T4 timer value represents the idle time allocated for the remote DTE station.
 - The T3 timer is defined on the X25.MCH statement. The T3 timer is valid only if STATION=DCE has been defined for the applicable MCH. This keyword is optional. The T3 timer value must be greater than the TPTIMER value.
 - When the T3 timer runs out, an RR command is sent to the DTE with the poll bit on. If no RR frame response is received, the RR command is resent. After all retries, if a RR frame response is not received, NPSI initializes a data-link resetting procedure by sending an SABM command to the DTE. If all attempts to reset the link have failed, and if no UA frame response is received, the link level is considered disconnected, and NPSI informs the upper layer using an INOP MCH station.

Packet Level

The V3R4 enhancements that allow NPSI to more closely comply with ISO 8208 and CCITT 1988 at the packet level include:

- **TOA/NPI Checking:** The A-bit (bit 8 of the GFI) is tested to determine the type of address block that is used in the Call Setup and Clearing packets. The following is a list of Call Setup and Clearing packets received by NPSI:
 - Incoming Call
 - Call Connected
 - Clear Indication
 - Clear Confirmation

Two formats are possible for the address block, with both formats distinguished by the A-bit (set to either 0 or 1).

- NPSI supports only non-TOA/NPI address formats, and always sends Call setup and Clearing packets with the A-bit set to 0.
- When the A-bit is set to 1, the TOA/NPI address format is used. For V3R4 and later, NPSI does not support the TOA/NPI address format, and does not send a Call Setup or Clearing packet with the A-bit set to 1. NPSI checks the A-bit for call setup and rejects all packets with the A-bit set to 1 by clearing the virtual circuit with diagnostic code X'42'.

Note: Clearing packets are processed as in previous releases.

- **V3R4:** CCITT 1988 has defined three new diagnostic codes:
 - 53 (X'35'): Time expired for call redirection
 - 78 (X'4E'): Maximum number of call redirections
 - 84 (X'54'): Network user identification (NUI) problem

Compliance with ISO Standards 7776 and 8208 (V3R5 Enhancements)

This section describes how NPSI V3R5 complies with ISO standards 7776 and 8208.

Link Level

The LAC layer of NPSI has been upgraded to conform more closely with ISO 7776.

Packet Level

The V3R5 enhancements that allow NPSI to more closely comply with ISO 8208 at the packet level include:

- New processing of the Clear Indication packet
 - NPSI V3R5 accepts only the following facility codes in Clear Indication packets:
 - X'08'
 - X'C1'
 - X'C2'
 - X'C5'

These facility codes are accepted without any verification of the facility requested during call setup.

- When NPSI receives a Clear Indication packet with a facility code that is not valid, NPSI sends back a Clear Request with the diagnostic code X'40' (Call Clearing Problem).
- New processing of the Clear Confirmation packet
 - NPSI V3R5 accepts only the following facility codes in Clear Confirmation packets:
 - X'C1'
 - X'C2'
 - X'C5'

These facility codes are accepted without any verification of the facility requested during call setup time or when a Clear Request is received.

- When NPSI receives a Clear Confirmation packet with facility code that is not valid, NPSI sends back a Clear Request with the diagnostic code X'40' (Call Clearing Problem).

Compliance with CCITT 1992 Standard

The optional facility field of a Call Requested packet has been extended by CCITT 1992 to allow for a maximum value of 255 bytes (510 digits). NPSI supports this extension with the added OPTFAC2 and OPTFAC3 keywords on the X25.OUFT statement.

X.25 ODLC Compliance with ITU-T 1993 Standard

X.25 ODLC provides support for the following:

- ISO8208 DTE to DTE protocol
- TOA/NPI addresses.

X.25 ODLC enforces the CCITT rules regarding ordering of virtual circuits. The range of logical channels must be as follows:

LCN	
0	
1)
) Permanent virtual circuits
)
LIC)
) One way incoming
HIC)
LTC)
) Two way
HTC)
LOC)
) One way outgoing
HOC)
4096	

LCN Logical Channel Number
 LIC Lowest Incoming Channel
 HIC Highest Incoming Channel
 LTC Lowest Two-way Channel
 HTC Highest Two-way Channel
 LOC Lowest Outgoing Channel
 HOC Highest Outgoing Channel

Within each type of VC, LCNs must be used consecutively.

Planning for NPSI

This section describes the considerations for NPSI that you should be familiar with when planning your installation:

- Software and hardware compatibility
- Storage requirements
- High performance transmission subsystem (HPTSS)
- 3746 Model 900 Connectivity Subsystem
- Connection without a PSDN
- X.25/ISDN interworking connection
- Logical link control types

Software and Hardware Compatibility

The following describes the software compatibility for NPSI:

Operating System	MVS, VM, and VSE operating systems.
Access Method	NPSI works with the same access methods as the corequisite NCP version. See <i>Planning for NetView, NCP, and VTAM</i> for more information.
Host Resident Programs	Programs that communicate with non-SNA destinations are either specifically written for the NPSI environment (for example, General Teleprocessing Monitor for Open Systems Interconnection (GTMOSI), Communication Subsystem for Interconnection (CSFI), or OSI Communication Subsystem) or have special support for start-stop ASCII devices (for example, CICS, IMS, TSO, and VM). All of these programs have modified support for LU type 1.

NPSI supports the IBM 3720 and the IBM 3745 communication controllers.

V3R8 and Later: NPSI V3R8 also supports X.25 connections through a 3746 Model 900 attached to the 3745.

For further information about hardware compatibility with NPSI, see *X.25 NPSI Version 3 General Information*.

NPSI allows swapping of the physical circuit to controller lines reserved for backup support. Also, the NCP load module can be loaded from the disk on the communication controller.

Storage Requirements

Because NPSI works in a communication controller under the NCP, plan your generation definitions to include NPSI storage requirements. NPSI has the NCP storage requirements as shown in Table 3-2 on page 3-19.

Table 3-2. NPSI Storage Requirements

NPSI Release	Executable Program Requires:	Each Virtual Circuit Requires:	Each Physical Circuit Requires:
V3R3	70K to 140K bytes	1.2K bytes	2K bytes
V3R4	90K to 280K bytes	1.3K bytes	2.1K bytes
V3R5	93K to 283K bytes	1.3K bytes	2.1K bytes
V3R6	98K to 290K bytes	1.4K bytes	2.1K bytes
V3R7	103K to 295K bytes	1.4K bytes	2.1K bytes
V3R8 and Later	118K to 327K bytes	1.4K bytes	2.2K bytes

V3R4 and Later: NPSI has moved most of the control blocks above the 4 mega-byte boundary to allow for the definition of larger networks.

HPTSS (NPSI V3R5 and Later)

NPSI V3R5 and later releases support the HPTSS duplex SDLC adapter that is dedicated to controlling one of two high-speed telecommunication lines (up to 2048 Kbps). The HPTSS line adapter provides a high-speed connection for the IBM primary ISDN gateway.

HPTSS uses the same modem interface management as X.21 leased or V35. X.21 switched interface is not supported. Although two lines can be linked on a HPTSS, only one can be activated at a time.

3746 Model 900 Connectivity Subsystem (NPSI V3R8 and Later)

The 3746 Model 900 CSS is an expansion frame that extends the connectivity of the IBM 3745 Communication Controller. All functions of NPSI, except interworking (see X25.MCH definition statement), are supported on a 3745 through a 3746 Model 900 attached. The following list describes the additional connectivity with the 3746 Model 900 attached:

- Physical interfaces
 - X.21 (384 Kbps or 1500 Kbps)

Note: X.21 leased and switched connections are supported on LIC type 11 and type 12. For LIC type 11, the minimum speed is 600 bps for leased connections and 56 Kbps for switched connections. For LIC type 12, the minimum speed is 56 Kbps. The LIC type 11 or type 12 hardware used with X.21 switched connections adapts to the speed supported by the ISDN terminal adapter.
- Up to 4 lines with frame level (LAPB) modulo 128 can be defined on a 64-address range. This range covers a communication link processor (CLP) with up to 2 LICs. Up to 8 modulo 128 lines can be defined on a CLP with 4 LICs attached (128-address range).

Note: Other lines operating with other protocols can share the same CLP and LICs with X.25 lines. See Table 3-3 on page 3-20 for recommended limits for lines sharing a CLP with X.25 modulo 128 lines.

Table 3-3. Recommended Limit for Number of Lines Sharing a CLP with X.25 Modulo 128 Lines

Number of X.25 Modulo 128 Lines	0	1	2	3	4
Number of X.25 Modulo 8, SDLC, or Frame-Relay Lines	120	90	60	30	0
Minimum Number LICs Required	4	4	3	2	1

Note: For a CLP with LICs using two consecutive 64-address ranges (128-address range), the maximum number of X.25 modulo 128 lines is 8. In this particular situation, care must be taken concerning the traffic handled by the CLP. Another major parameter to consider is the frame level window size, which can limit the number of non-acknowledged frames transmitted to a number lower than modulo 128 allows (127). When a CLP enters congestion status, X.25 frames are discarded. If this discard mechanism is entered too frequently, it means that the CLP is overloaded. An additional CLP might be necessary to handle the excess traffic.

Connection without a PSDN

NPSI generally uses a PSDN to communicate with a remote DTE, but using a PSDN is *not* a requirement. NPSI can communicate with a destination node without going through a PSDN.

Note: A PSDN is no longer required when using a QLLC subarea node in conjunction with the switched virtual circuit subarea communication (SVCSC) function.

The destination node can be attached through the following:

- Direct connection (communication controller clocks the physical circuit)
- Modem eliminator
- Regular data circuit

Two cases must be distinguished:

Case 1: NPSI functions as a DCE.

The link level in NPSI acts as a DCE station. Primary and secondary addresses are reversed. DCE to DTE commands carry the address 03, while responses carry the address 01. NPSI does not send a RR packet with poll commands when activity does not exist on the line.

The packet level in NPSI works as a DCE station with the following differences:

- In Call packets, facilities are processed as in a DTE.
- Cause and diagnostic codes are set as in a DTE, except when CAUSE=DCE is coded on the X25.MCH statement. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information about the codes.

Case 2: The remote device operates as a DCE. In this case, NPSI operates as a DTE.

Note: For clarity, the NPSI library is written so that NPSI appears to operate as a DTE. Note that the NPSI MCH can also act as a DCE with the previous restriction.

X.25/ISDN Interworking Connection

X.25/ISDN interworking allows a host application to communicate with a remote terminal by establishing an X.25 virtual circuit session through an ISDN and a PSDN by a B-channel. Both incoming connections, initiated by a remote terminal (an X.25 network subscriber), and outgoing connections, initiated by NPSI acting as the DTE, are provided.

The IBM 7820 Terminal Adapter (TA) is used to establish the B-channel connection. Line speed on the connection is 64 Kbps.

For information about the support of packet mode equipment by an ISDN, see CCITT Recommendation X.31, Case A. For information about the interface between a DTE and a DCE for terminals operating in the packet mode and accessing a PSDN through ISDN, see CCITT Recommendation X.32.

Configuration

The physical lines of NPSI (MCH lines) are leased resources. In the configuration described in this section and shown in Figure 3-1 on page 3-22, the physical interface between MCH lines and the TA/7820 can be either a V.35 or an X.21 leased interface. The PSDN entity managing the ISDN switched path (in/out) is called either the interworking access unit (IAU) or simply the access unit (AU).

Establishing the Physical Path

Establishing a physical path as defined in Recommendation X.32 depends on which party initiates the ISDN connection. When NPSI (DTE) initiates the ISDN connection, the operation is called *dial-in-by-DTE*. When the PSDN (on remote terminal request) initiates the ISDN connection, the operation is called *dial-out-by-PSDN*. Each MCH line of NPSI that allows interworking connections is dedicated to either dial-in or dial-out switch access operations (not both) by system generation parameters.

- To dedicate an MCH line to interworking dial-in operations, code the interworking keyword on the X25.MCH statement (INTWLINK) as follows:

```
INTWLINK=DIALDTE
```

- To dedicate an MCH line to interworking dial-out operations, code INTWLINK as follows:

```
INTWLINK=DIALPSDN
```

Establishing LAPB and Virtual Circuits

The NPSI DTE conforms to the nonidentified DTE service offered by the PSDN, as described in Recommendation X.32:

- NPSI performs no identification.
- The X.25 subscription set is specified as *network default*.
- Assignment of logical channels is by network default:
 - Only one virtual-call logical channel with dial-out operation
 - One or more virtual-logical channels with dial-in operation

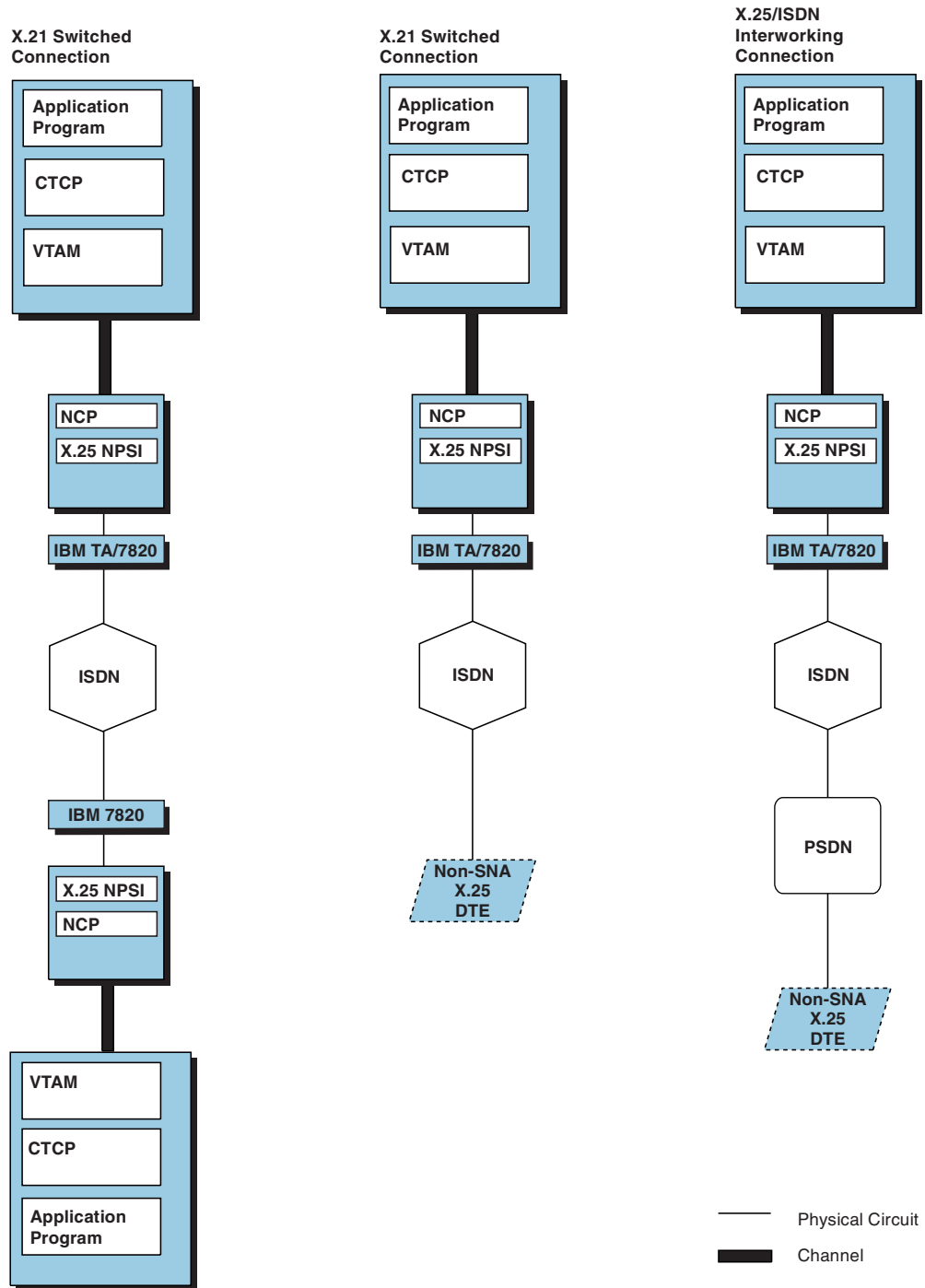


Figure 3-1. Configuration Examples Supported by X.25 NPSI and TA

- The direction of virtual call placement is related to the direction of the dial operation, as shown in Table 3-4.

Table 3-4. Direction of a Virtual Call for the Dial Operation

Dial Operation and Service Used	Capability for DTE Originating and Receiving Virtual Calls
Dial-in-by-DTE	Originating virtual calls
Dial-out-by-PSDN	Receiving virtual calls

- NPSI does not perform dynamic link address assignment depending on the direction of the call. Addresses are assigned at system generation, which means that two separate MCH lines are required for support of dial-in and dial-out operations:
 - An MCH line dedicated by the INTWLINK=DIALDTE and STATION=DTE keywords to dial-in operations. In this case, the access unit acts as a DCE for link-layer addressing.
 - An MCH line dedicated by the INTWLINK=DIALPSDN keyword to dial-out operations. In this case, when the access unit acts as a DTE, define the MCH line at system generation with STATION=DCE. When the access unit acts as a DCE, define the MCH line at system generation with STATION=DTE.
- The DTE should not be permitted to make paid calls or to receive reverse-charged calls.

Disconnecting the Physical Path

The way the PSDN access unit disconnects the switched access path depends on which of the two dial operations is involved:

- For dial-in operations, the PSDN access unit disconnects the switched-access path only in the absence of any virtual call during a specified time period (T14).
- For dial-out operations, the PSDN access unit disconnects the switched-access path as soon as the virtual circuit is released.

Functions Supported by NPSI on Interworking Connections

Table 3-5. Functions Supported by NPSI on Interworking Connections

	PCNE (LLC type 0)	PSH (LLC type 2)	QLLC (LLC type 3)	GATE (LLC type 4)	PAD (LLC type 5)	DATE
MCH dedicated to dial-in	Yes	Yes	Yes*	No	Yes	No
MCH dedicated to dial-out	Yes	Yes	Yes*	Yes	Yes	No

Note: Do not define the interworking MCH line as a complex unit. PHYSRES=YES is not allowed.

X.21 Leased Connections

For X.21 leased connections or lines that use a speed greater than 19 200 bps, the scanner (TSS) or the 3746 Model 900 must have knowledge of the line speed to select the LIC type 4B mode of operation for the TSS or the LIC type 11 or 12 mode of operation for the 3746 Model 900.

For leased lines using speeds greater than 19 200 bps, you must define the following keywords on the X25.MCH statement:

- X21INTWK=OLD or 1984
- SPEED=xxxx

X.21 Switched Connections (NPSI V3R6 and Later)

NPSI provides you with support for X.21 switched connections. X.21 switched connections allow for DTE-to-DTE communication using a terminal adapter across an integrated services digital network (ISDN). ISDN is a set of standards that provides for advanced high-speed communications and allows the simultaneous transmission of voice and data across the same line.

X.21 switched connections allow you to use ISDN services through NPSI. The physical resource associated with an X.21 switched connection is a GATE MCH that is dedicated to the ISDN connection.

V3R6 and Earlier: Only one CTCP, such as the Open Systems Interconnection (OSI) Communication Subsystem, can be associated with this MCH. The connection is through an IBM 7820 terminal adapter. Figure 3-1 on page 3-22 shows the X.21 switched configurations supported by NPSI and the IBM 7820.

V3R7 and Later: Multiple CTCPs can be associated with this MCH.

V3R8 and Later: X.21 switched access to ISDN through the IBM 3746 Model 900 is allowed. Figure 3-2 on page 3-25 shows an X.21 switched configuration through the 3746 Model 900 and a Hitachi Terminal Adapter Model HN-5110-24.

During system generation, define the following keywords on the X25.MCH statement to provide for X.21 switched connections:

Keyword	Result
X21SW	<p>Specifies that the physical line is an X.21 switched line. When you specify X21SW=YES or X21SW=T70DXE, you must code the X25.MCH statement with the following keywords:</p> <pre data-bbox="695 1136 972 1413">X21NTWK=1984 (or OLD) GATE=GENERAL LLCLIST=LLC4 CONNECT=NO LCN0=NOTUSED SUBADDR=NO FTPI=NO DIRECT=NO SPEED=xxxx</pre> <p>Note: Specify SPEED=xxxx to allow for the selection of LIC type 4B for the 3745 without the 3746 Model 900 attached when the speed is greater than 19 200 bps.</p> <p>V3R8 and Later: Specifying SPEED=xxxx is mandatory for the 3745 with the 3746 Model 900 attached to allow selection of LIC type 12. Refer to the description of the SPEED keyword in “X25.MCH Statement” on page 5-45 for more information.</p>
X21NTWK	<p>The keyword X21NTWK on the X25.MCH statement specifies whether the physical line is attached to an X.21 network. At system generation, the value used for SPEED is processed and passed to the scanner at link activation only when X21NTWK=OLD or X21NTWK=1984 is specified on the X25.MCH statement.</p>

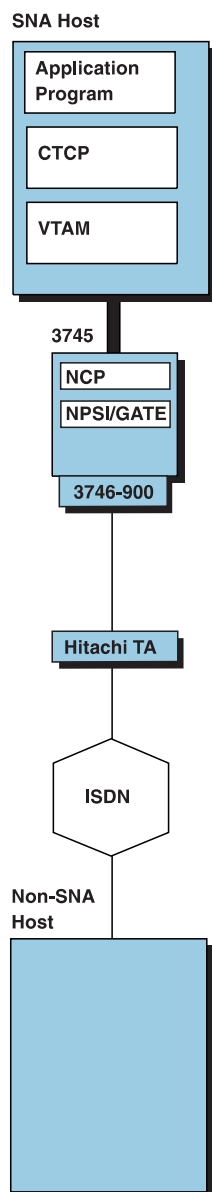


Figure 3-2. ISDN Configuration Example with the IBM 3746 Model 900 and a Hitachi Terminal Adapter

- X21RTYTO Specifies the timeout interval that the scanner waits before retrying a call-out.
- X21RTYCT Specifies the number of retries that are performed by the scanner following a timeout.
- X21INACT Specifies an inactivity timer, which checks for an active virtual circuit on the MCH. If no activity is found during the specified time, the X.21 connection is cleared.

The MCH defined for X.21 switched connections is dedicated to X.21 switched access. You can only define switched virtual circuits on this MCH.

Logical Link Control (LLC) Types

NPSI supports five LLC types.

The LLC type used depends on the type of device with which NPSI communicates, and the characteristics of the communication.

X.25 ODLC supports only LLC type 3.

LLC types define the support functions required. LLCs can be classified either as SNA to SNA connections or as SNA to non-SNA connections.

- SNA to SNA connections:
 - LLC type 2 connects to SNA peripheral node DTEs through a PSH interface.
 - LLC type 3 connects to SNA peripheral nodes or SNA subarea nodes using qualified logical link control (QLLC).
- SNA to non-SNA connections:
 - LLC type 0 connects to X.25 non-SNA DTEs.
 - LLC type 4 connects to non-SNA DTEs that use GATE support.
 - LLC type 5 connects to non-SNA DTEs through a PAD.

Note: LLC type 1 is not used.

The following sections provide a detailed description of the LLC types and DATE.

SNA Connections

LLC types 2 and 3, which connect to SNA destinations, are explained in this section.

Logical Link Control Type 2: LLC type 2 connects the host to an SNA DTE by using a physical services header (PSH). NPSI processes the SDLC commands through the PSH.

To create an LLC type 2:

- PVC: Use the X25.LINE, X25.PU, and one or more X25.LU statements; do not use the X25.VC statement.
- SVC: Use the X25.VC statement or the X25.LINE and X25.PU statements together.

See Chapter 5, “Resource Definition Reference,” for more information about the X25.VC, X25.LINE, X25.PU, and X25.LU statements.

The following are examples of type 2 connections:

- 3174 cluster controller connected to the PSDN through a Network Interface Adapter (NIA)
- 3174 connected to the PSDN using PSH microcode support

Figure 3-3 on page 3-27 illustrates an LLC type 2 connection.

Logical Link Control Type 3: LLC type 3 connects the host with SNA DTEs using qualified logical link control (QLLC). Packets with the Q-bit contain SDLC commands. Packets without the Q-bit contain Information frames (I-frames).

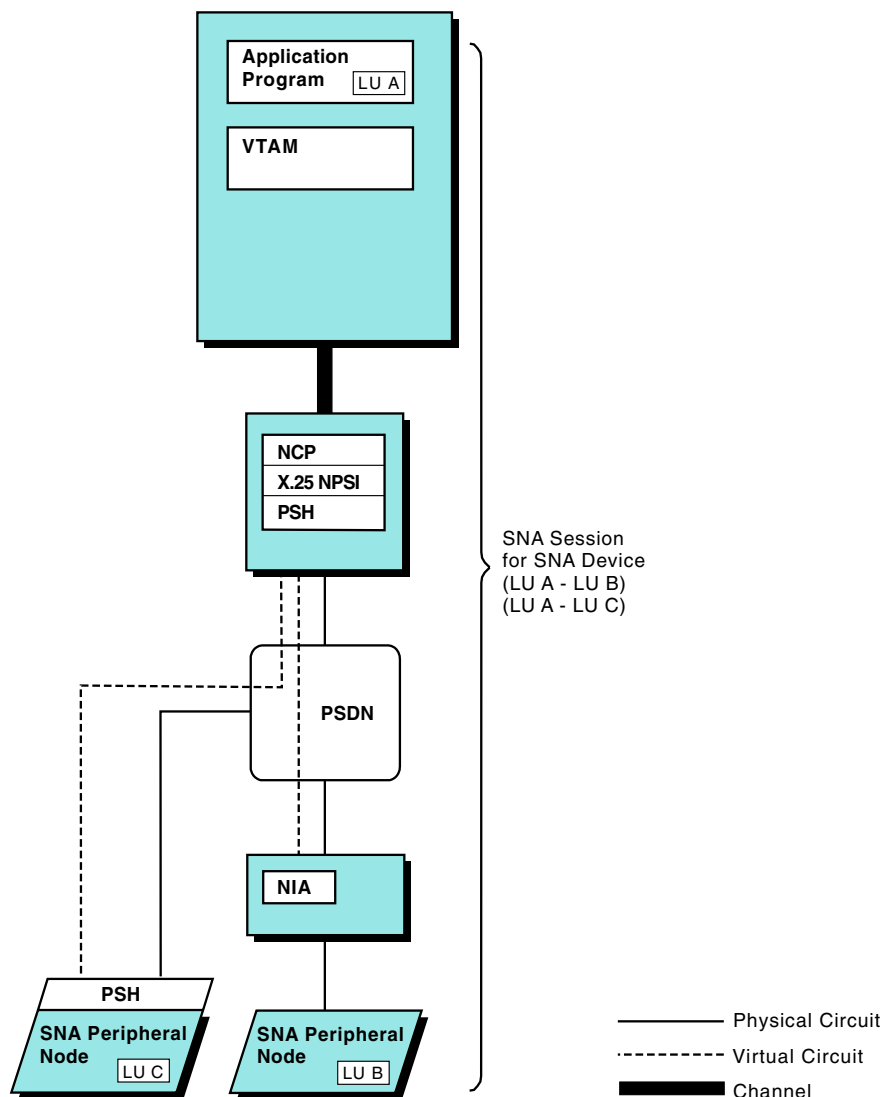


Figure 3-3. Logical Link Control Type 2

The peripheral node QLLC supports the SNA type 2.1 node. This type of node allows peer-to-peer sessions to be set up between peripheral nodes. Subarea node QLLC supports the SVCSC function and the short hold mode (SHM). SVCSC allows subarea-to-subarea communication using a switched virtual circuit.

Note: A PSDN is not required when you use the SVCSC function.

QLLC also supports a primary SNA type 2.1 peripheral node. In this case, the NCP acts as a secondary node. The enhanced SNA type 2.1 support function (casual connection) requires the secondary side of NPSI to be V3R3 or a later release. The primary side must be NPSI V3R4 or a later release. X.25 ODLC is capable of acting as either the primary or secondary side.

Figure 3-4 on page 3-29 shows an LLC type 3 that uses peripheral node QLLC support to attach to a peripheral node. Figure 3-5 on page 3-30 shows an LLC type 3 that uses subarea network node QLLC support to attach to a communication controller using NPSI.

To create an LLC type 3:

- Peripheral node
 - PVC: Use the X25.LINE, X25.PU, and one or more X25.LU statements; do not use the X25.VC statement.
 - SVC: Use the X25.VC statement or the X25.LINE and X25.PU statements together.
- Subarea node
 - PVC: Use the X25.LINE and X25.PU statements together; do not use the X25.LU statement.
 - SVC: Use the X25.LINE and the X25.PU statements to define the SNA resources corresponding to the NCP LINE and PU to the remote subarea. Then use the X25.VC statement to define the virtual circuit.

The following examples are LLC type 3 connections:

- SNA host connected to the PSDN through NPSI or X.25 ODLC
- SNA device with integrated X.25 support connected to the PSDN
- SNA host connected to an SNA DTE through an X.25 network
- SNA type 2.1 node connected to another SNA type 2.1 node through NPSI or X.25 ODLC and a PSDN

See Chapter 5, “Resource Definition Reference,” for more information about the X25.VC, X25.LINE, X25.PU, and X25.LU statements.

NPSI and X.25 ODLC allow you to activate, load, and dump a remote NCP in different ways. You can use an SVCSC connection to activate a remote NCP with NPSI and X.25 ODLC. After a remote NCP with NPSI has been activated, you can use an SVC or PVC to transfer a load module from the host to a remote communication controller’s disk.

NPSI and X.25 ODLC also allow you to use an SVC or a PVC to transmit an initial program load (IPL) request from the host or disk to an active, remote communication controller. The IPL request causes a load module that resides on the 3720 or 3745 (with or without the 3746 Model 900 attached) communication controller’s disk to be loaded into the communication controller. However, you cannot initially load a remote communication controller over an SVC or PVC.

The following are required:

- Switched major nodes when switched X.25 lines are used
- The following keywords on the X25.LINE statement:
 - MONLINK=YES
 - IPL=YES
 - SWMMTID=*name*

NCP verifies that both MONLINK=YES and IPL=YES are valid specifications for a switched or leased subarea link.

SWMMTID=*name* contains the name of a switched SMMF table defined by a group of SWMMT statements. SWMMT is a statement that is valid only for switched subarea lines.

See Chapter 5, “Resource Definition Reference,” for more information about the X25.VC, X25.LINE, X25.PU, and X25.LU statements.

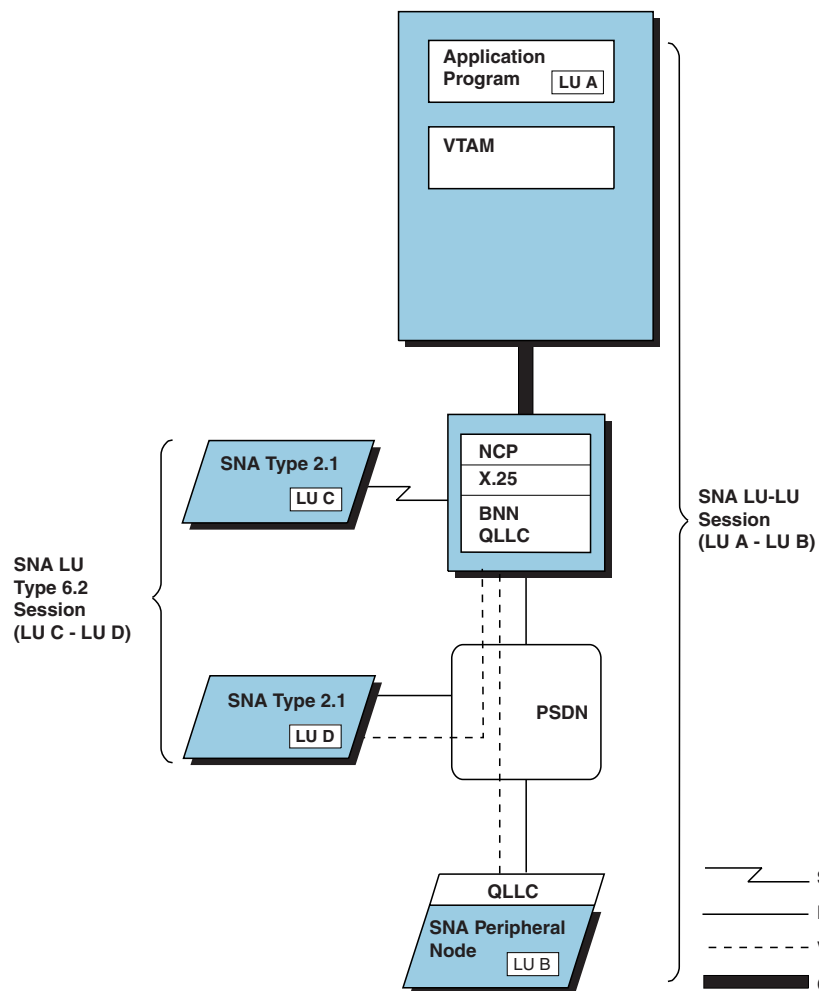


Figure 3-4. Logical Link Control Type 3 (Peripheral Node). The X.25 connections shown can be provided by the NPSI licensed program or, when the IBM 3746 Model 900 is used, by NCP X.25 ODLIC.

NPSI and X.25 ODLIC also provide the capability for a host-initiated dump to disk by means of a host-initiated forced abend. The abend causes a dump to disk and a re-IPL of the load module. Once the remote communications controller has been activated, the dump can be transmitted from the disk to a host across a PVC or an SVC.

Non-SNA Connections

This section explains the LLC types 0, 4, and 5, which connect to non-SNA destinations.

NPSI communicates with non-SNA destinations by simulating an SNA physical unit type 1 and logical unit type 1 for each non-SNA device. This simulator allows the SNA host and its associated applications to communicate with the non-SNA devices. The simulation function is called the LU simulator (LUSIM). The LU simulator converts outgoing SNA requests from the host into X.25 packets, and converts the X.25 packets returning from the non-SNA DTE into SNA requests.

LUSIM facilitates host control over the non-SNA destination. Examples of this control include clearing the virtual circuit and operating a dial-out connection.

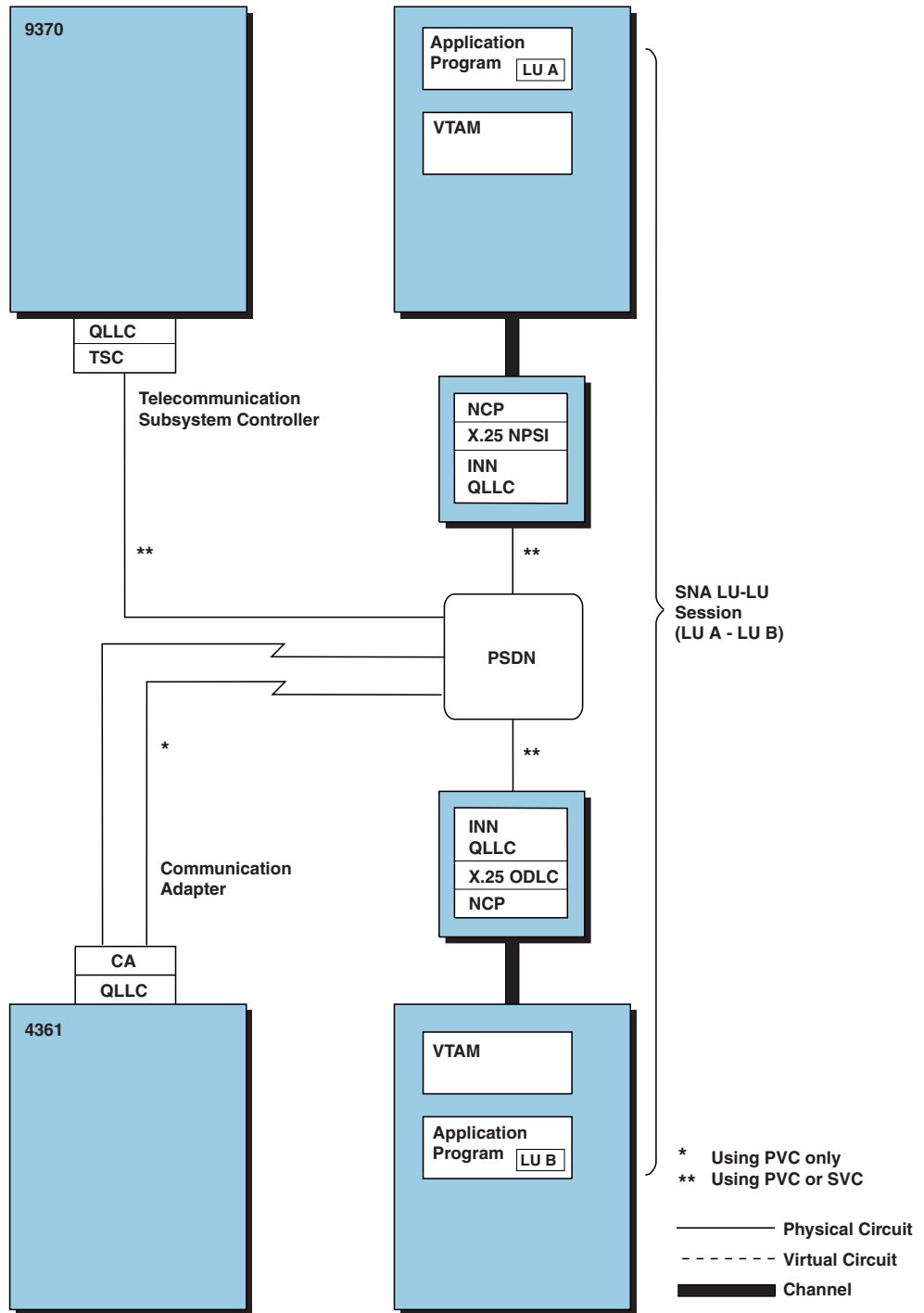


Figure 3-5. Logical Link Control Type 3 (Subarea Node). The X.25 connections shown can be provided by the NPSI licensed program or, when the IBM 3746 Model 900 is used, by NCP X.25 ODLC.

Logical Link Control Type 0: LLC type 0 (PCNE) connects the host with non-SNA DTEs that use X.25.

To create an LLC type 0:

- PVC: Use the X25.VC statement or the X25.LINE, X25.LU, and X25.PU statements together.

- SVC: Use the X25.VC statement or the X25.LINE and X25.PU statements together.

See Chapter 5, “Resource Definition Reference,” for more details on using the X25.VC, X25.LINE, X25.LU, and X25.PU statements.

The following are examples of LLC type 0 connections:

- Non-SNA host connected to the PSDN
- Non-SNA device connected to the PSDN without PAD

See the LLC type 0 example shown in Figure 3-6 on page 3-32.

Logical Link Control Type 4: Type 4 LLCs are designed to allow users complete control over their virtual circuit connection through the use of a feature called general access to X.25 transport extension (GATE). The GATE support function is based on PCNE or LLC type 0.

In an LLC type 0 connection, NPSI handles all the X.25 control packets automatically. However, for a type 4 connection, a host application called the CTCP must be written to control the virtual circuit.

The CTCP controls the setup and termination of the virtual circuit by exchanging commands with NPSI GATE. In addition, the CTCP processes all X.25 control and data packets. Command codes are contained in the first byte of each request unit exchanged between the CTCP and NPSI. Because all application data passes through the CTCP, the GATE CTCP is often called a *relay* program. GATE converts incoming packets into PIUs. Then, GATE adds the appropriate 1-byte command code to the request unit, depending on the type of packet received, before sending the PIUs to the CTCP. Upon receipt of the request unit, the CTCP interprets the command code and acts upon the command.

GATE converts outgoing PIUs into packets. The packet type is determined by the first byte of the outgoing request or response unit (RU).

The CTCP communicates with GATE by using one of two types of LU-LU sessions. The first type of session is a control session. The control session is between the CTCP and one of the LUs for the physical circuit used by the LLC type 4. Its main use is to convey call set up and diagnostic commands. The second type of session is a data session. The data session is between the CTCP LU and the LU for the virtual circuit. It is used to convey commands and data after the connection has been established. For further information, see *X.25 NPSI Version 3 Host Programming*.

LUs for the physical circuit are created according to the number of CTCPs implied by the keywords on the X25.MCH statement. Virtual circuits of a given MCH can communicate with up to 28 CTCPs. The GATE keyword on the X25.MCH statement must be coded as GATE=GENERAL.

To create an LLC type 4:

- PVC: Use the X25.VC statement or the X25.LINE, X25.PU, and X25.LU statements together.
- SVC: Use the X25.VC statement or the X25.LINE and X25.PU statements together.

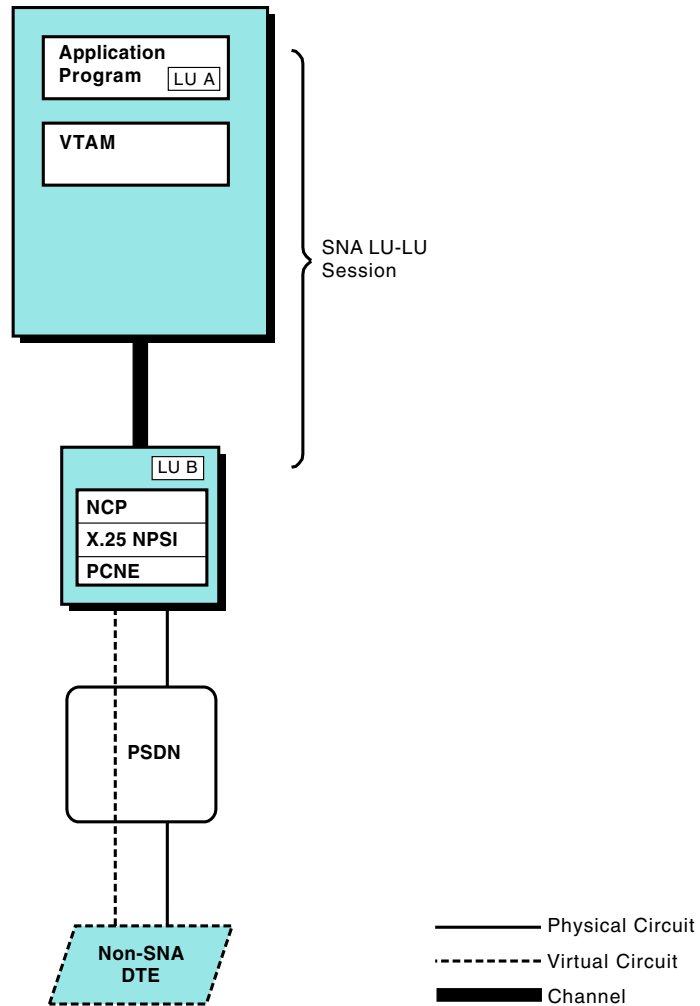


Figure 3-6. Logical Link Control Type 0

See Chapter 5, “Resource Definition Reference,” for further descriptions of the X25.MCH and X25.VC statements.

An example of an LLC type 4 connection is one to GTMOSI, CSFI, or OSI/Communication subsystem. In certain countries, IBM can provide these CTCPs for use with GATE. The CTCP is connected to a videotex terminal, or a start-stop device through GATE and PAD. See the example of GATE shown in Figure 3-7 on page 3-33.

Fast Connect Option for GATE: NPSI includes an option for GATE virtual circuits called fast connect. Fast connect provides a quick way to connect non-SNA terminals to an SNA host processor. Rather than going through SNA session establishment for each call, fast connect saves time during call establishment by linking X.25 resources with preestablished SNA sessions. After SNA session establishment is complete, the sessions remain active when the X.25 virtual circuits are cleared and when the MCH fails or is deactivated.

SNA resources, used to map fast connect switched virtual circuits, are declared as nonswitched to NCP and the access method.

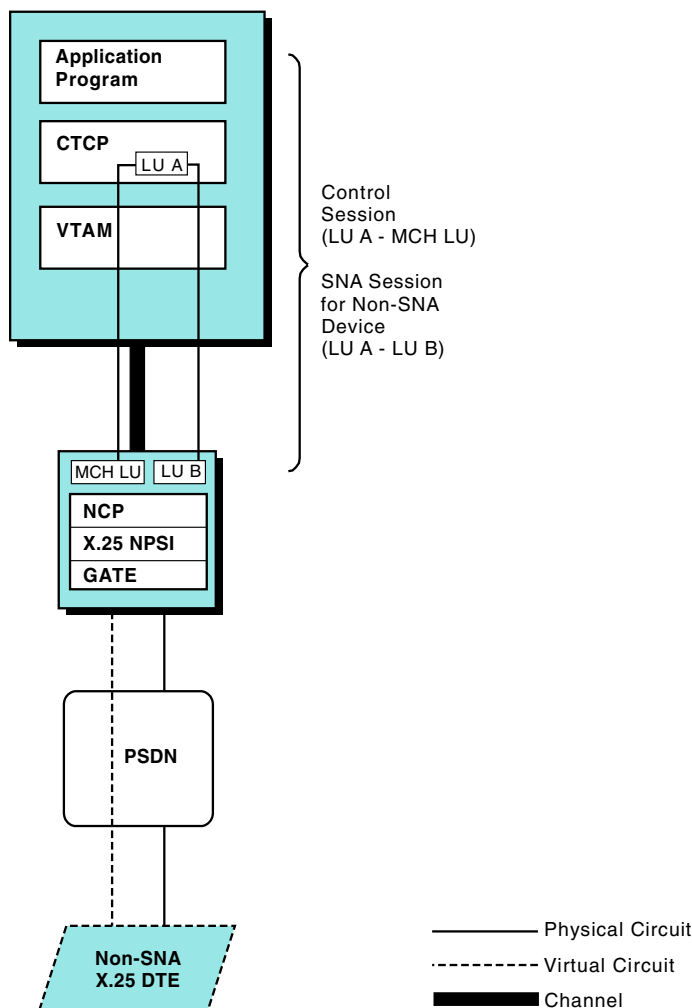


Figure 3-7. Logical Link Control Type 4 (GATE)

As with the GATE function, two types of sessions are established. The difference between the sessions is that only diagnostic commands flow on the control session. All other commands, including call setup commands, flow on the data session.

When fast connect is selected on an MCH, the entire MCH is dedicated to the fast connect operation and cannot be used for other LLC types.

At system initialization, all available SNA sessions between the CTCP and NPSI are opened by VTAM's automatic logon (LOGAPPL). NPSI fast connect then assigns each incoming call to one of the preestablished sessions, thereby eliminating all VTAM processing associated with SNA session establishment. After the user activity is complete, the virtual circuit is cleared. However, the associated SNA session is maintained in a ready state to receive any subsequent calls on any virtual circuit.

Fast connect is useful when there is a heavy demand for quick connections of a short duration. You can use fast connect when you have:

- A single CTCP for each MCH
- A single CTCP for multiple MCHs
- Multiple CTCPs connected to the same MCH

Planning

For example, you might use fast connect in a system that connects videotex terminals through a PAD to an SNA host processor.

Example 1: If there is only one CTCP, or if the peak traffic period for each CTCP occurs at the same time, you can choose to connect only one CTCP for each physical circuit. Figure 3-8 shows an example of such a configuration.

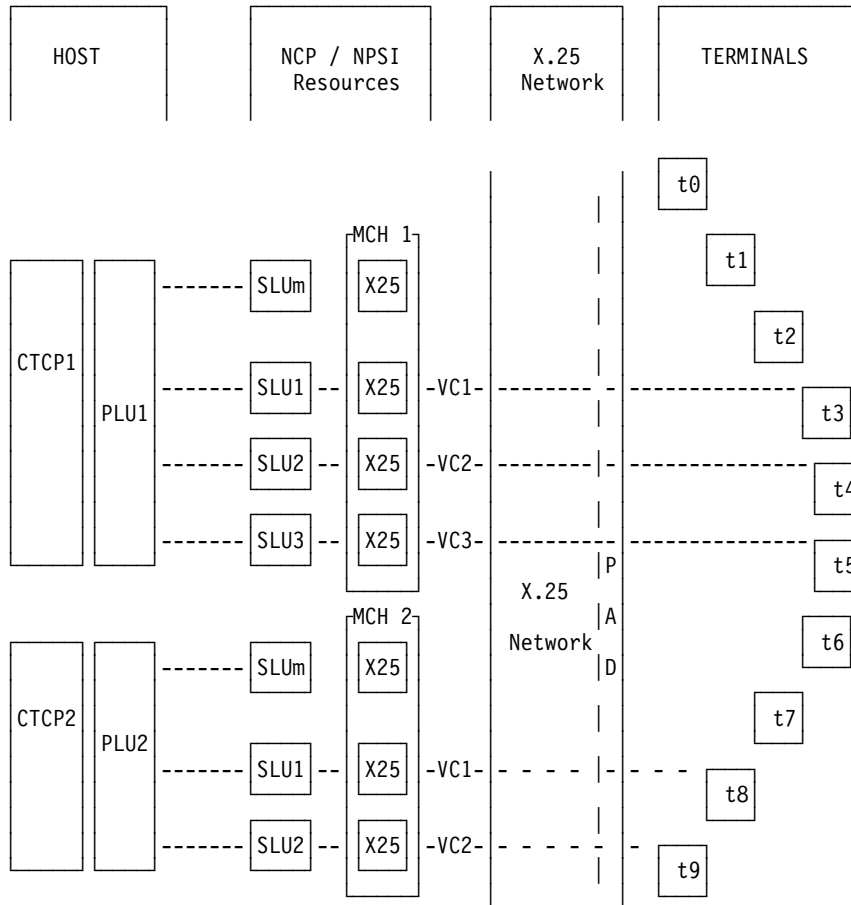


Figure 3-8. Fast Connect with One CTCP for One Physical Circuit

Example 2: If peak traffic periods occur at different times on different CTCPs, the set of virtual circuits for one physical circuit can be shared by several CTCPs. Each virtual circuit can connect to only one CTCP at a time. In this case, more than one CTCP is defined for each physical circuit as shown in Figure 3-9.

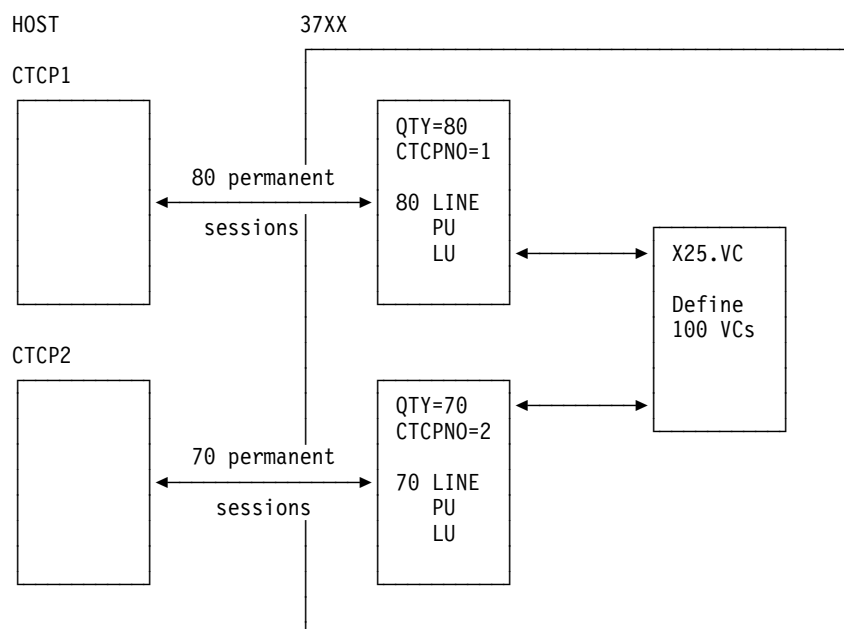


Figure 3-9. Fast Connect with Two CTCPs for One Physical Circuit

As shown in Figure 3-9, for example, CTCP1 has a peak time from 4:00 p.m. to 5:00 p.m. CTCP2 has a peak time from 7:00 p.m. to 8:00 p.m. CTCP1 can communicate with 80 SNA resources. CTCP2 can communicate with 70 SNA resources. The physical circuit contains 100 virtual circuits.

During peak time for CTCP1, up to 80 virtual circuits can be used for CTCP1. Concurrent with this, CTCP2 handles the remaining 20 virtual circuits.

The reverse occurs during peak time for CTCP2. Up to 70 virtual circuits can be used by CTCP2, which lets CTCP1 use up to 30 virtual circuits.

To generate fast connect, code GATE=GENERAL on the X25.MCH statement. You also need to use the CONNECT keyword with a value of either YES, CUD0, or SUBD. If you code YES, do not code the CUD0 or the SUBD keywords, because there is only one CTCP for each MCH. If you code CUD0 or SUBD, you must code the corresponding CTCP keyword.

You can also code the following keywords of the X25.MCH statement:

- LUNAME
- LOGAPPL
- MODETAB
- VMODTAB
- TAXUNIT

Note: The valid values for the TAXUNIT keyword are 0, 32, 64, or 128, with 0 being the default. When you specify 0 on this keyword, NPSI counts the number of packets that are sent and received.

If you are connecting the physical circuit to multiple CTCPs, code an X25.FCG statement for each CTCP that is in communication with the MCH. An X25.FCG statement creates LINE, PU, and LU statements for fast connect resources.

Use X25.VC to describe X.25 resources associated with fast connect virtual circuits. The CALL keyword must be coded consistently with your PSDN subscription. TYPE=S must be coded and X25.VC must be used to describe the virtual circuits. X25.LINE cannot be used. The order of the statements must be:

```
X25.MCH    CONNECT=SUBD or CUD0,CTCP=
X25.LCG
X25.VC     LCN=(n1,n2),TYPE=S
X25.FCG    QTY=m1,CTCPNO=x
X25.FCG    QTY=m2,CTCPNO=y
```

See Chapter 5, "Resource Definition Reference," for further descriptions of the X25.MCH, X25.FCG, and X25.VC statements. See Appendix B, "System Generation Examples," for examples of fast connect coding.

Fast Transaction Processing Interface: The Fast Transaction Processing Interface (FTPI) is a protocol between NPSI and a specialized GATE CTCP. The rules of this protocol are:

- Only one SNA LU-LU session is bound between one CTCP and NPSI. This session can contain multiplexed traffic from all of the virtual circuits connected to the CTCP.
- The CTCP session uses session parameters that minimize extra traffic.

Figure 3-10 shows the association of an SNA session with X.25 virtual circuits for FTPI.

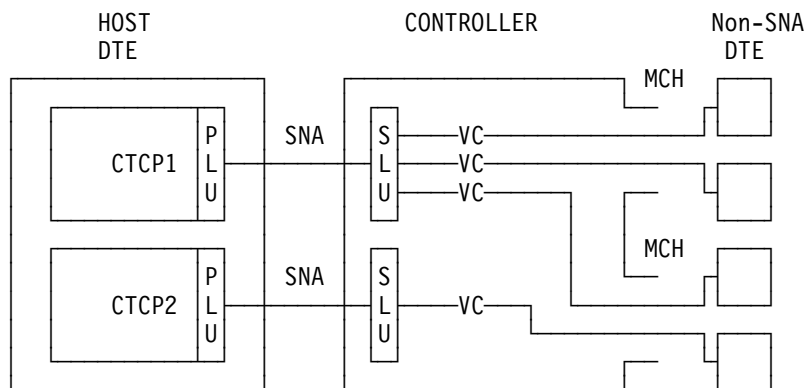


Figure 3-10. Association of SNA Session with X.25 Virtual Circuits for FTPI

When FTPI is selected on an MCH, the entire MCH is dedicated to the FTPI operation and cannot be used for other LLC types. Both SVCs and PVCs are supported on an FTPI MCH.

V3R7 and Earlier: An FTPI MCH could not be defined with PVCs only.

V3R8 and Later: An FTPI MCH can be defined with PVCs only.

The objective of the FTPI protocol is to enhance the performance of the entire communication system provided by normal GATE processing. FTPI provides advantages for network throughput and utilization in both the host and communications processor.

The following are the advantages FTPI provides for network throughput:

- Eliminates the establishment of multiple SNA sessions between NPSI and the CTCP.

As in fast connect, FTPI has a preestablished SNA session between NPSI and the CTCP.

- Recommends use of a special SNA session bind image.

By using certain bind parameters, FTPI can increase the throughput for the SNA session. This increase results in greater throughput of the entire system.

The following session parameters are recommended for FTPI:

- LU type 1
- Half-duplex contention protocol
- Exception response only
- No bracketing
- No chaining
- No segmenting
- Use of fixed pacing

The purpose of these session parameters is to reduce the overhead in processing the data. Not using bracketing, chaining, or segmenting eliminates the processing overhead associated with performing these tasks.

- Uses exception timers to reduce adverse effects of low utilization.

Because FTPI multiplexes across the SNA session to the CTCP, if the line utilization is very low, the data waits to be sent to the host. FTPI uses exception timers to reduce the adverse effect of this low utilization.

The timers set a maximum time period for waiting to transmit data to the host. If the PIU is not filled at the expiration of this timer, the buffer is sent without filling. This expedites the transmission of data to the host and reduces extended time waiting for the buffer to fill.

- Provides load balancing.

When a CTCP is overloaded or in the process of being stopped, the CTCP can notify NPSI that it should no longer forward Incoming Call packets to that CTCP. Alternative CTCPs can be defined in a CTCP table. This table is utilized when the primary CTCP cannot be accessed or data is not being accepted.

When FTPI receives an Incoming Call packet, FTPI checks the session with the primary CTCP. If it cannot be accessed, the Incoming Call is forwarded to one of the available CTCPs in a round robin fashion.

FTPI support is indicated by coding the X25.CPL statement, and the FTPI keyword on the X25.MCH statement.

NPSI V3R7 and NCP V7R2 Only: NPSI APAR IR27307 must be applied for FTPI to work.

Logical Link Control Type 5: LLC type 5 connects the host with non-X.25, non-SNA DTEs that use an X.25 PAD. Type 5 connections are similar to type 0 connections. The exception is, with type 5, the host program (transparent option) or NPSI (integrated option) can send and receive qualified packets in addition to non-qualified packets. PAD service functions are controlled by means of qualified packets. PAD support is indicated by coding the PAD keyword in the X25.MCH statement.

To create an LLC type 5 virtual circuit:

- PVC: Use the X25.VC statement or the X25.LINE and X25.PU statements together. You must also code an X25.LU statement.
- SVC: Use the X25.VC statement or the X25.LINE and X25.PU statements together.

See Chapter 5, “Resource Definition Reference,” for details about using the X25.VC, X25.LINE, X25.PU, X25.LU, and X25.MCH statements.

The following examples are type 5 connections:

- Non-SNA, non-X.25 device connected to the PSDN using a PAD that supports CCITT Recommendations X.3, X.28, and X.29.
- Non-SNA, non-X.25 device connected to the PSDN using a nonstandard PAD.

Figure 3-11 on page 3-39 shows an LLC type 5 connected through integrated PAD support. Figure 3-12 on page 3-40 shows an LLC type 5 connected through transparent PAD support.

Note: Transparent PAD can also be used to control non-SNA X.25 remote DTEs that do not use a PAD. The application has control over qualified and unqualified data packets and Reset and Interrupt packets. Control over these packet types is usually a requirement for non-SNA X.25 DTEs offering PAD services that do not comply with CCITT Recommendations X.3, X.28, and X.29.

Virtual Circuits that Use DATE

NPSI provides extended support, called dedicated access to X.25 transport extension (DATE), for LLC types 0, 2, 3, and 5. You can use DATE for security control, resource allocation, accounting, and other extended requirements.

DATE support requires the use of a CTCP. Virtual circuits that use DATE support are set up and taken down by commands passed between the CTCP and NPSI. The CTCP controls the setup and the termination of the virtual circuit, and processes all X.25 control and qualified packets, except when using LLC type 5.

DATE converts incoming control packets into PIUs. DATE adds the appropriate 1-byte command code to the RU, before sending the PIU to the CTCP. Upon receipt of the RU, the CTCP interprets and acts upon the command code.

DATE then converts outgoing PIUs passed by the CTCP into control or qualified packets. Before the packet is sent to the remote DTE, DATE deletes the 1-byte command code, builds the control or qualified packets, and transmits the packets to the PSDN.

Communication with NPSI is performed using two types of LU-LU sessions. The first type of session is between the CTCP logical unit and the physical circuit logical unit. The second type is between the application logical unit and the virtual circuit logical unit. Once the virtual circuit is set up, the protocol depends on the type of virtual circuit. The CTCP is used during session setup and termination; it also handles control packets and qualified packets.

The application communicates directly with the terminal for data exchange. Consequently, the CTCP is not involved. For control and qualified packets, the CTCP communicates with the NPSI DATE function over the CTCP to MCH LU session. The third byte of the RU is used to indicate the packet type.

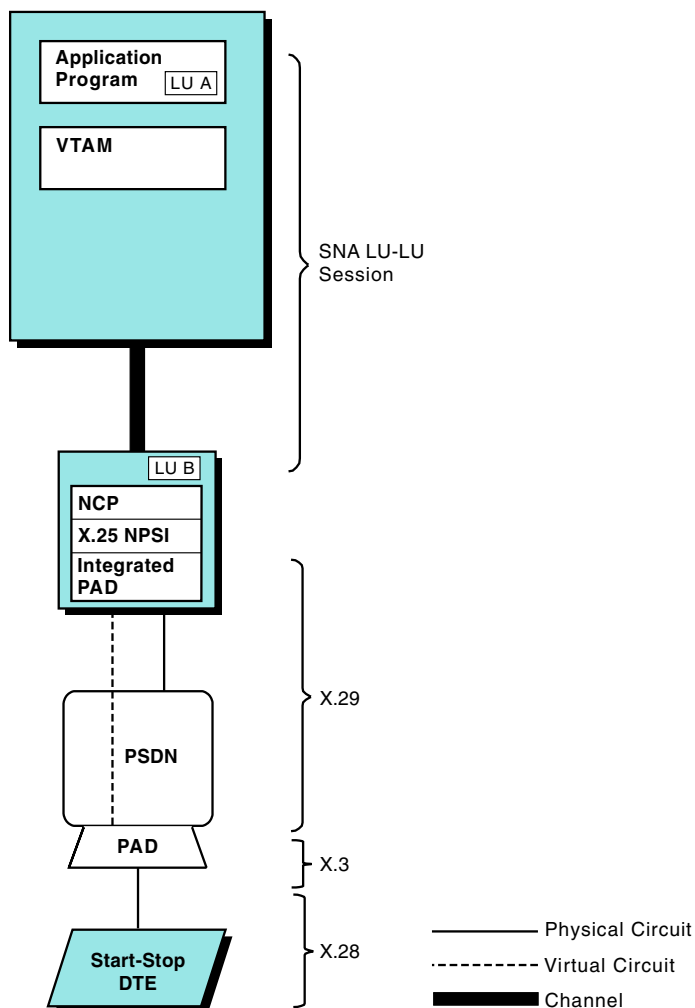


Figure 3-11. Logical Link Control Type 5 (Integrated PAD)

The following examples are DATE-type connections:

- Non-SNA device connected to a PSDN through a nonstandard PAD
- Nonstandard network connected to an SNA host

To generate DATE support, first code GATE=DEDICAT in the X25.MCH statements, and then for:

- PVC: Use the X25.LINE or X25.VC statement. X25.PU and X25.LU statements must be coded together with the X25.LINE statement.
- SVC: Code the CALL keyword consistent with your PSDN subscription in the X25.VC or X25.LINE statement. the X25.PU statement must be coded with the X25.LINE statement.

GATE and DATE cannot use the same physical circuit. SVCSC is not supported on a DATE MCH.

For LLC type 5, DATE provides support for either transparent *or* integrated PAD on the same MCH.

See Chapter 5, “Resource Definition Reference,” for detailed information about the X25.MCH, X25.LINE, X25.VC, X25.PU, and X25.LU statements.

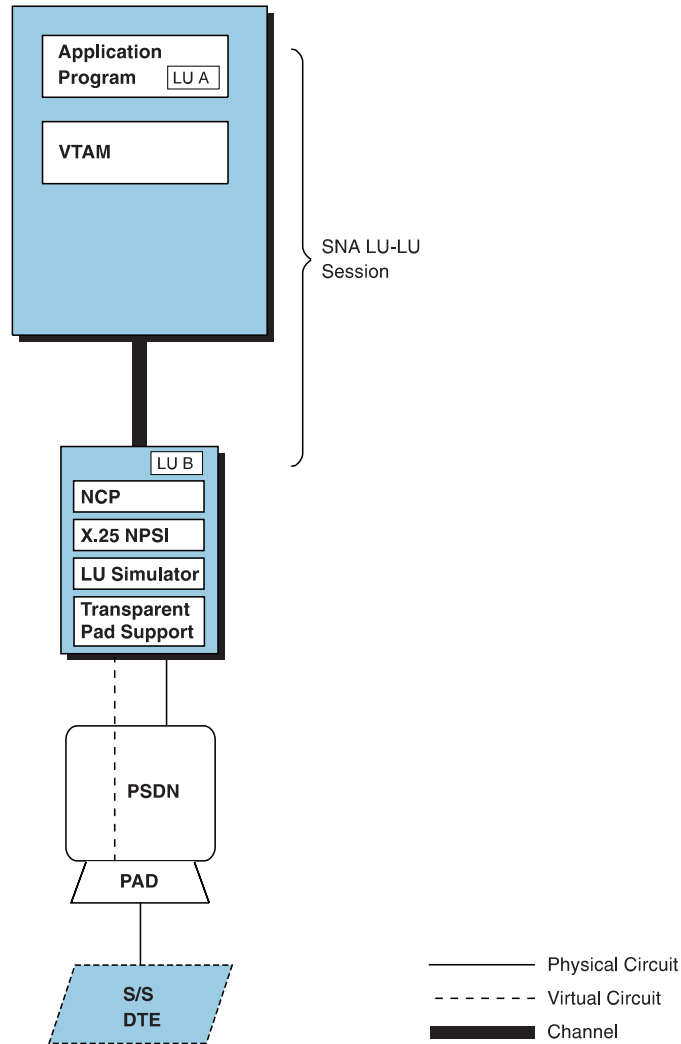


Figure 3-12. Logical Link Control Type 5 (Transparent PAD)

Figure 3-13 on page 3-41 shows how DATE works with the CTCP to control the virtual circuit.

Logical Link Control Type Selection

The LLC type for PVCs is selected through the use of the LLC keyword in the X25.LINE statements or the X25.VC statements associated with the PVCs. The LLC type is defined for PVCs during the generation process. (The LLC type for SVCs is not defined at generation. During generation, the LLCLIST keyword of the X25.MCH statement specifies the virtual circuit types allowed on this MCH. The type for any given session is selected during virtual circuit setup.)

X.25 ODLC supports LLC selection using CUD0 and only LLC3 is allowed.

During call-in, NPSI decides the LLC type to use by one of the following methods:

- Call user data byte 0 (CUD0)

An LLC can be selected using IBM-defined standard CUD0 values. If these standard values do not fit your needs, NPSI allows you to create a correspondence table to specify user-defined CUD0 fields. Through this table, devices

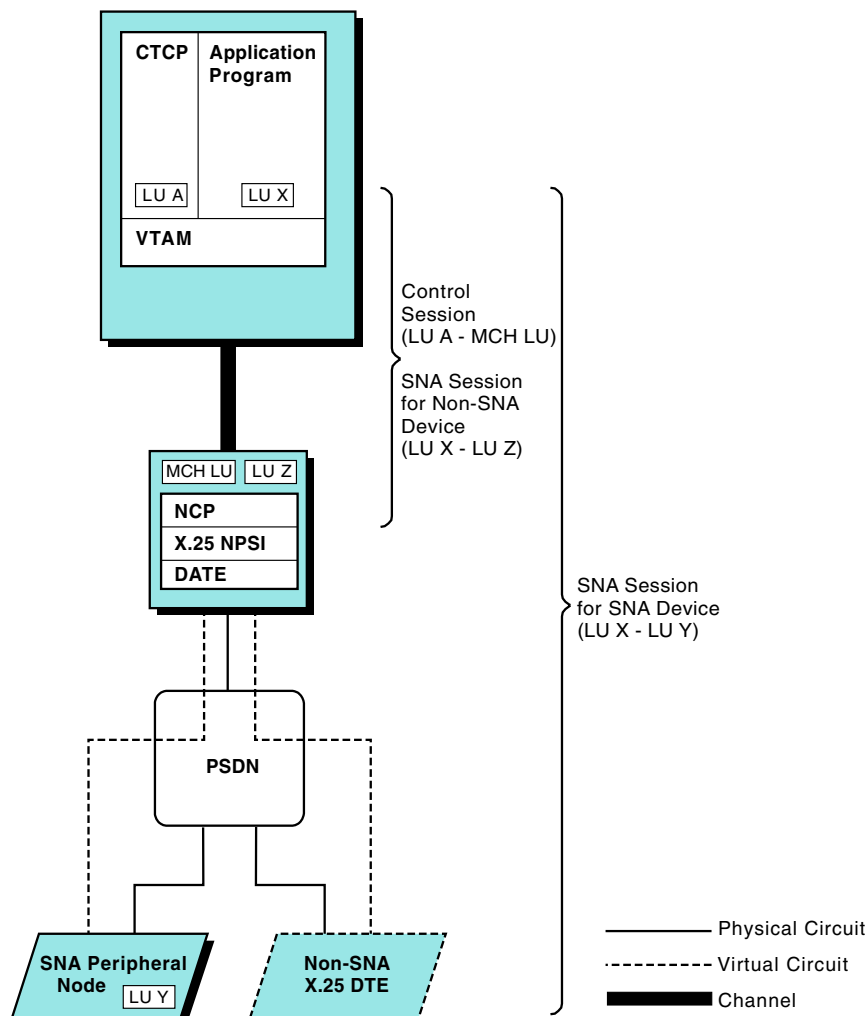


Figure 3-13. Virtual Circuits That Use DATE

and software that do not use the IBM-defined standard CUD0 values can select LLC types.

- Subaddressing

The last digit in the subaddressing field is used to select an LLC. The subaddressing field is the last digit at the end of the called DTE address.

- CALL ACCEPTED command (if DATE is used)

In response to the Incoming Call packet, the CALL ACCEPTED command containing the LLC type is sent from the CTCP to the logical unit associated with the physical circuit.

During call-out, the LLC type is selected through VTAM. For virtual circuit types 0, 2, 3, and 5 that do not use DATE, code the type in the L character of the DIALNO or DLCADDR keywords in the VTAM PATH statement. For call-outs that use DATE, the CALL REQUEST command from the CTCP to the DATE function of NPSI specifies the LLC. For GATE, LLC type 4 is automatically selected by NPSI when a GATE CTCP makes an outgoing call.

NPSI Support of RU Chaining for Long Non-SNA Messages

During session establishment, the VTAM BIND parameters specify the maximum size of the request unit (RU) that one logical unit can send to another logical unit.

Note: The maximum RU size must be greater than the X.25 packet size, and lower than the X25.MAXPIU value coded on the BUILD statement. If not, the BIND is rejected with a negative response parameter error (sense X'1005').

To send a message containing more information than will fit into one RU, logical units divide this information into a series of requests. This series is called a chain, and each RU in the chain is defined as one of the following:

- First-in-chain (FIC)
- Middle-in-chain (MIC)
- Last-in-chain (LIC)
- Only-in-chain (OIC)

NPSI either ignores the SNA chaining information and processes each chained element independently or, if you code MBITCHN=YES on the X25.MCH statement at system generation, converts these SNA chains to packet sequences using a combination of two bits: the M-bit and the D-bit. These bits determine how to map RU chains into packet sequences and packet sequences into RU chains. NPSI can create two types of packet sequences:

- Complete packet sequence (CPS)
- M-bit sequence

A CPS contains contiguous full data packets, with the M-bit set to 1 and the D-bit set to 0, followed by any other data packet.

An M-bit sequence contains a CPS series. Each packet within the series has the M-bit set to 1, except for the last packet of the last CPS.

Inbound Flow

Usually, NPSI considers each inbound message from a non-SNA DTE a CPS and converts it into an OIC RU. However, NPSI also provides RU chaining through the MBITCHN keyword on the X25.MCH statement. If you specify both MBITCHN=YES and DBIT=YES, a CPS series is converted into an RU chain. The length of the RU chain is determined by the length of the CPS.

For NPSI to support the delivery confirmation bit, you must specify a definite response protocol in the BIND request.

Outbound Flow

NPSI processes outbound PIUs destined to non-SNA DTEs as only-in-chains, unless you specify MBITCHN=YES and DBIT=YES on the X25.MCH statement. In this case, each chain of RUs is converted into a CPS series that is linked together using the M bit. This series is then sent to the non-SNA DTE as a chain.

If you specify MBITCHN=YES and DBIT=NO, a chain of RUs (FIC, MIC, and LIC) is converted into a single CPS. The CPS is sent to the non-SNA DTE.

Quality of Service Facility

This section describes the quality of service (QOS) facility for NPSI.

NPSI Support for QOS Facility

NPSI's support for QOS allows you to prioritize the traffic on one or more MCH virtual circuits. QOS enhances the internal NPSI process for queuing a packet before transmission to the network for multiple peripheral node traffic priorities.

You can define this priority either at the call setup phase on a SVC using the QOS facility, or at installation time on PVCs using the DEFPRI keyword at system generation.

Because the QOS priority of data is an OSI concept and requires an upper layer to select the priority, the QOS negotiation applies only to GATE SVCs. For the other users, the priority is always taken from the DEFPRI keyword at system generation.

Note: The priority service requested by OSI at the network level is handled on a virtual circuit basis, and only regulates the transmission to the network, not to the flow inside NPSI.

Priority Definition

QOS applies only to peripheral node traffic, because subarea traffic always receives a high priority.

The QOS facility is defined as an end-to-end service that is passed unchanged between participating DTEs.

The QOS facility is composed of the following three subpriorities:

- The priority of data on connection
- The priority to gain a connection
- The priority to keep a connection

Of the three subpriorities, only the priority of data on a connection is supported by NPSI.

Range Definition

The range of specified values used over the NPSI network are from 0 (lowest priority) through 14 (highest priority). The numbers 15 through 254 (X'0F'—X'FE') indicate a reserved value, and the number 255 (X'FF') indicates an unspecified value. The OSI priority of data is split into the following value ranges:

- Low priority = 0—3 (X'00'—X'03')
- Normal priority = 4—7 (X'04'—X'07')
- High1 priority = 8—11 (X'08'—X'0B')
- High2 priority = 12—14 (X'0C'—X'0E')
- Reserved = 15—254 (X'0F'—X'FE')
- Unspecified = 255 (X'FF')

If the negotiated value is reserved or unspecified, the default priority value specified on the X25.MCH statement is used.

Piggybacking Techniques

Piggybacking means a received frame or packet is acknowledged in the next outgoing frame or packet, rather than in an isolated Receive Ready (RR) frame or packet. Increased piggybacking occurs if the packet or link level window size is greater than 2.

Link Level

The degree of piggybacking is increased by delaying the RR transmission for a short period of time. This delay allows additional information frames (I frames) to arrive in the outbound queue. The acknowledgment is piggybacked on an outbound I frame that uses the same physical circuit.

The RR transmission time delay is defined by the T2 timer. The T2 timer is calculated using the two values defined in the X25.MCH statement:

- SPEED
- T1TIMER

For the link-level piggybacking to occur, the T1TIMER must be coded on the X25.MCH statement. The SPEED keyword defines the line speed of the physical circuit. The T1TIMER keyword defines the maximum time the DCE will wait for a frame acknowledgment.

When service is continuous (24 hours a day, 7 days a week), and the MCH speed must be altered without changing the generation, code SPEED and T1TIMER keywords that will work with both SPEEDs and T1TIMERS. By specifying the SPEED and T1TIMER keyword values that generate the smallest T2, you can benefit from link-level piggybacking in both cases. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for computation of T2.

Packet Level

Unless the receive packet window is about to close, an RR packet is not sent to acknowledge a received packet. Instead, the acknowledgment is made by the next packet that is sent. Packet-level piggybacking occurs only if the packet window size is greater than two.

A system generation option, PLPIGGYB on the X25.MCH statement, allows you the option of turning off packet-level piggybacking, even if the window size is greater than two. In this case, every received data packet is acknowledged with an RR packet. However, using the PLPIGGYB keyword on the X25.MCH statement may inhibit optimum performance.

Planning for NPSI and X.25 ODLC in the Network

This section describes NPSI planning considerations for the overall network configuration. NPSI resides in a communication controller under the control of NCP. X.25 ODLC is a function of NCP. Both access a host and network resources through an access method, such as VTAM. Statements and keywords in VTAM and NCP that specifically affect X.25 connections are described in Chapter 6, “Coding NCP and VTAM Parameters for X.25 Support.”

Other factors that affect your network planning for NPSI include:

- SNA considerations
- Host application program considerations
- Network resource recovery considerations

SNA Considerations

Because NPSI and X.25 ODLC are used to connect SNA equipment through an X.25 interface, the following SNA considerations affect X.25 planning:

- PIU segmentation
- 3270 printer LU type selection
- Pacing parameters

PIU Segmentation

Two parameters determine how the user data is segmented for outbound transmissions from the host:

- NCP buffer size (BFRS) specified in the NCP BUILD statement
- MAXDATA keyword on the PU statement

NCP Buffer Size: The optimal buffer size for the support of SNA peripheral nodes is not the same for NPSI and NCP. If LLC type 2 is not used, an appropriate value is 124; if LLC type 2 is used, the value 120 reduces the number of excess packets. Set the optimal buffer size to a value that avoids the creation of small packets, which require as much processing as large packets.

MAXDATA Keyword: This keyword specifies the maximum number of bytes in a PIU. These parameters can be tuned to reduce the number of packets flowing across the DTE and DCE interface. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information.

3270 Printer LU Type Selection

You should use 3270 printers in SNA character string (SCS) mode LU type 1 rather than in data stream compatibility (DSC) mode LU type 3.

Pacing Parameters

To reduce unnecessary data transfer and pacing delays, you should either specify no pacing or set the pacing counts to the maximum allowed by device buffering. Consult the device component description manual to determine the pacing parameters. An isolated pacing response causes as much CCU utilization as a full data packet.

Host Application Program Considerations

When planning for NPSI, consider the following host application requirements:

- Non-SNA communication
- Transparent PAD implementation
- CICS, IMS, and TSO adjustments

Non-SNA DTE Communication

You must synchronize the two end points of the X.25 connection. Synchronization means that the end points manage data formats and application requirements in a compatible manner.

To synchronize end points, make sure that one or both ends know the configuration of the data that the partner is sending. The configuration includes:

- Character set (ASCII, EBCDIC, or other)
- Interface parity
- Data format (information layout within the data stream)
- Control characters and their associated effects
- Application control

You can discover these details by familiarizing yourself with the end points. If the end point that you want to communicate with is not at your site, call the remote location for the necessary information.

Transparent PAD Implementation

The host application program using transparent PAD support adds a control character to the Request unit in the first byte. This control byte specifies one of the following six message types:

- Data packet without Q-bit
- Data packet with Q-bit
- Interrupt request
- Interrupt confirmation
- Reset packet
- Reset confirmation

See *X.25 NPSI Version 3 Host Programming* for more information about the programming requirements for Transparent PAD.

CICS, IMS, and TSO Adjustments

CICS, IMS, and TSO are commonly used applications. To communicate with the simulated LU, you may need to make adjustments to these applications. The host application program must consider such items as bracketing and presentation.

You might want to design the application program to reduce the required dialogs between the host and remote user. To do this, use formatted screens or modified data fields.

General Adjustments: To enable CICS, IMS, and TSO applications to communicate with a logical unit simulated by NPSI, consider the following:

- Bracketing

See *X.25 NPSI Version 3 Host Programming* for more information about bracketing considerations.

- Presentation level
Presentation level includes character set used, communication parity, and screen layout.
- Application level
Application level includes application responses, nature of the application behavior, and conversation termination requirements.
- BIND parameters
CICS and IMS can create their own BIND parameters rather than using the session parameters from the VTAM logmode entry, such as, RUSIZE and PACING. Check the application's terminal for specific parameter information.
- Definite or exception response
The use of definite or exception response is defined within the BIND parameters and can be created in these subsystems. A definite response ensures message integrity, but can have a disastrous effect on performance in an X.25 interface when used on an LLC type 0, 4, or 5 with the D-bit. Whenever possible, you should use exception response in these cases; however, the application requirements or hardware support can be a constraint to this specification. See *VTAM Programming* and *X.25 NPSI Version 3 Host Programming* for more information.

CICS Adjustments: In CICS, consider the following:

- Definite response can be requested by the:
 - MSGINTEG keyword on the DFHPCT macro
 - DEFRESP parameter on the SEND or CONVERSE commands
- The WAIT option on the SEND command can cause additional RUs to be sent just for change-of-direction or end-bracket indicators. The WAIT option can slow down response time by increasing the number of transmitted packets.

IMS Adjustments: In IMS, response mode can be more efficient than non-response mode. Consider an IMS fast path that automatically uses response mode.

TSO Adjustments: In TSO, the application level requires special consideration because TSO does not include the recovery facilities available in CICS and IMS. Only line mode operation is available when LLC type 0 or 5 is used.

Network Resource Recovery Considerations

Session continuation allows LU-LU sessions to remain active when the owning system service control point (SSCP) fails. With the owning SSCP inoperable, an alternate SSCP can acquire ownership of the resource. When the original SSCP becomes operable, it can regain ownership. This is true for PVCs and SVCs attached to either SNA or non-SNA remote DTEs. When using virtual circuits that are under DATE, you need to observe special considerations. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information.

NetView Performance Monitor

The NetView Performance Monitor (NPM) allows you to manage performance and accounting data that is exclusive to the SNA portion of the network.

NPM already supports accounting and performance data collection for all SNA LUs (MCH and virtual circuit).

You must modify your system generation to allow the following functions:

- Process commands coming from NPM through NCP
- Collect performance and accounting data
- Store performance data in a collect record provided by NCP
- Report accounting PIUs to NPM through NCP

Accounting

You can use NPM to check the billing of your network subscribers through the accounting data collected for the X.25 resources.

Accounting data is provided for the PU associated with a virtual circuit:

- At connection establishment and clearing for SVCs
- At MCH activation and deactivation for PVCs
- Each time certain accounting counters exceed a system-generated threshold

If you want to keep accounting information for X.25 resources, you must specify `SESSACC=YES` on the `BUILD` statement. You must also specify a byte and segment threshold for reporting purposes. The `X25.BYTTHROD` and `X25.SEGTHROD` keywords of the `BUILD` statement specify these parameters, respectively. You must also activate accounting by specifying the `ACCOUNT` keyword of the `X25.MCH` statement. This keyword has three options:

- NO
- YES
- A list of LLCs that should have accounting support

The `NPADTEAD` keyword is required on the `X25.MCH` statement, and provides the identifier of the MCH for NPM purposes. This keyword can be used to specify the X.121 address for the connection, but the contents of the keyword are not verified.

The `UACCTNG` keyword on the `GENEND` statement defaults to the correct module name(s). This specification is added only if at least one `ACCOUNT` keyword is coded for an X.25 resource.

Performance

You can use NPM to check performance characteristics and assist in tuning the network subscription through the performance data collected for the X.25 resources.

The NPACOLL keyword on the X25.MCH statement specifies the levels at which performance data is kept. The specification is:

```
NPACOLL = (MCHLINE, MCHPU, VCPU)
          (NO, NO, NO)
```

where:

MCHLINE	Collects performance data at the frame level.
MCHPU	Collects global performance data at the packet level.
VCPU	Collects individual virtual circuit performance data at the packet level.
NO	Data is not kept at the level. NO is the default.

See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information about the performance data that is supported and how to use the data to tune your network.

Planning for NPSI and X.25 ODLC Migration

This section includes information about generation migration for NPSI and X.25 ODLC. SSP includes support for migration of NPSI statements and keywords, provided that the NPSI statements are imbedded in the NCP definition that is to be migrated. For more information about migration, see the *NCP V7R8.1 Migration Guide*.

Migration from NPSI V1 to NPSI V3 Releases

Verify that you have modified the current NPSI source statements to allow creation of a valid load module and access method definition. Be sure to:

- Modify all NPSI statements.

All version 1 macros must be changed to include a period between the X25 portion and the suffix. For example, the X25MCH macro becomes the X25.MCH statement.

- Remove the X25BUILD statement.

If NDF finds an X25BUILD statement, it will be flagged as an error and a load module will not be generated until it is removed. Many keywords previously coded on X25BUILD are not required now that NPSI is generated under NDF. Others have been moved to the NCP BUILD statement.

- Modify NPSI keywords to allow NDF and VTAM processing.

Several NPSI keywords must be modified. For example, the IDNUMH keyword previously coded in X25BUILD must be coded as X25.IDNUMH and placed on the NCP BUILD statement.

Also code USERGEN=X25NPSI. The USERGEN keyword on the NCP OPTION statement indicates the name of the load module that processes the NPSI generation statements in NDF.

Code NEWDEFN=YES so that NDF builds a new generation definition for input into VTAM.

- Place the NPSI statements after all SS and BSC definitions, and before SNI statements that define the nonnative networks (NETWORK, GWNAU, and PATH).

- Modify the LCGDEF keyword.

Change LCGDEF=a(b) to LGGDEF=(a,b) to permit NDF processing. If you need to place more than one pair, use the following format:

```
LCGDEF=((a,b),(c,d),..., (y,z))
```

- Note of the change in the DSTNODE default value.

The default value of the DSTNODE keyword on the X25.LINE statement is now BNN (peripheral node) for both PVCs and SVCs.

- Modify the fast connect virtual circuit definitions.

Fast connect virtual circuits must be declared as SVCs. Previously, they were declared as PVCs in the fast connect programming request for price quotation (PRPQ).

- Modify the fast connect CTCP.

Formerly, call-outs in the fast connect PRPQ were directed to the MCH LU. However, now they must be directed to the SNA resources that are used to map SVCs.

- Code SPEED and T1TIMER on the X25.MCH statement.

To obtain maximum piggybacking, specify the SPEED and the T1TIMER keywords on the X25.MCH statement.

- Note of the change in the X25.VC and the X25.LINE of the CALL keyword for the DATE MCH. This is now coded according to the PSDN subscription.
- Note that the CUD field for LLC type 3 now only contains X'C3'. The bytes that follow can be added as needed by using the USRFILD keyword of the X25.OUFT statement.
- The ZZZZZ field of the DIALNO keyword of the PATH statement in the SMN must now be coded in all cases where NPSI is to generate a Call Request for LLC type 0 or LLC type 5.
- Note that the X25.USGTIER keyword must be coded on the BUILD statement.
- Eliminate X25.END statement keywords.

If NDF finds keywords coded on the X25.END statement, NDF creates a warning message and suppresses the creation of the load module.

- Note that the NSTDFSC keyword on the X25.NET statement is required for NPSI to accept nonstandard facilities in the Incoming Call packet.

Migration from NPSI V3R2 to NPSI V3R3

To migrate from V3R2 to V3R3, consider the following:

- The enhanced SNA type 2.1 boundary support function (casual connection) requires one NPSI V3R3 for secondary support. For primary support, you can use either NPSI V3R2 or V3R3.
- The IDBLK definitions within VTAM switched major nodes may need to be changed depending on the value chosen for the IDBLKC, IDBLKG, and IDBLKP keywords of the X25.MCH statement.
- The IDNUM definitions within VTAM switched major nodes may need to be changed to reflect the change in the use of the X25.IDNUMH keyword on the NCP BUILD statement.
- The X25.IDNUMH keyword in the NCP BUILD statement specifies only the first hexadecimal digit of the IDNUM value for non-SNA connections rather than the first two hexadecimal digits.
- The support for RU chaining for long non-SNA messages function is now a generation option on the X25.MCH statement.
- NETTYPE=4 must be specified on the X25.NET statement for networks that do not allow duplicate facilities.
- The remote CCU loading enhancement is available when both NPSIs are at the V3R3 level.
- The CRAFTRC keyword must be replaced by the RESETINO keyword on the X25.NET statements during system generation.

Migration from NPSI V3R3 to NPSI V3R4

To migrate from V3R3 to V3R4, consider the following:

- In conformance with Recommendation X.25 from CCITT 1988, NPSI requires that the DCE support a T3 system inactivity timer.
- NPSI also tests the A-bit (bit 8 of the general format identifier) to determine the type of Address Block that is used in the Call Setup packet, accepting all A-bits set to 0, and rejecting all A-bits set to 1.

The DTE checks the length of the following packets:

- Restart Confirmation
- Restart Indication
- Incoming Call
- Clear Request
- Clear Confirmation
- Reset

Packets that are too long are flagged as having format errors.

The Clear and Reset packets have new diagnostic codes.

- The DM keyword becomes optional for V3R4.
- NPSI has the option of interacting with NPM.

V3R3 Only: NPACOLL=NO is automatically forced on the X25.LINE, X25.PU, and X25.VC statements, because NPSI did not support NPM.

V3R4 and Later: NPSI ignores the NPACOLL keyword on the X25.PU and X25.LU statements. NPACOLL is not valid on these statements.

- The LCNHI value coded on the LCGDEF keyword of the X25.MCH statement must also appear on a X25.VC or X25.LINE statement to define the last virtual circuit of a logical channel group.

V3R3 Only: NPSI supports the following as nonstandard facilities:

- QOS facility
- Called Address Extension (AEC)
- Called Line Address Modified Notification (LAMOD)

V3R4 and Later: NPSI supports the following as standard facilities:

- QOS facility
- Called AEC
- Called LAMOD

V3R4 and Later: NPSI must analyze the Call Accepted packet sent by the CTCP to extract the selected value of the QOS facility. This is a deviation from preceding NPSI releases, and is not the same process used for packet size and window size selection, where the selected values are in the CALL ACCEPTED command from the CTCP.

- The following options have been added to the RFAC keyword on the X25.NET statement:
 - Called Address Extension Facility (AEF)
 - Called Line Address Modified Notification Facility (LAMOD)
 - OSI priority facility of QOS negotiation facilities (OSIPRI)

Note: When LCNHI is not defined in the last X25.LINE or X25.VC statement of the group, the NDF process ends with the message BAL133I.

Migration from NPSI V3R4 to NPSI V3R5

To migrate from NPSI V3R4 to V3R5, consider the following:

Calling DTE Address Verification

If your current VTAM level does not perform verification, or if VERID is not coded in your system, VTAM ignores the CV25 control vector appended by NPSI V3R5 to the REQCONT.

NCP V6R1 Currency

For delayed logical link activation, if you are using an early level of VTAM, or if the MCH is owned by a different SSCP than the virtual circuits, the virtual circuits can be activated before the MCH.

For users migrating from V3R4 to V3R5, mandatory keywords are not necessary. Users migrating from other releases should see “Migration from NPSI V1 to NPSI V3 Releases,” “Migration from NPSI V3R2 to NPSI V3R3,” or “Migration from NPSI V3R3 to NPSI V3R4” for intermediate migration adjustments.

Migration from NPSI V3R5 to NPSI V3R6

To migrate from NPSI V3R5 to V3R6, consider the following:

X.25/ISDN Interworking

In NPSI, you can dedicate an MCH to interworking connections by using the INTWLINK keyword on the X25.MCH statement. All LLCs are supported on an MCH dedicated to interworking connections.

Once an MCH is dedicated to interworking connections, no other type of connection can be defined for the MCH.

GATE MCH LU Processing

GATE CTCPs that issue the new CTCP command MLUNAME receive a negative response (X'080C', function not supported).

Migration from NPSI V3R6 to NPSI V3R7

Consider the following information when you migrate from NPSI V3R6 to V3R7:

- Modulo 128 is supported at frame level if you code MMODULO=128 on the X25.MCH statement. If you select modulo 128, the value range for MWINDOW is 1–127.
- Coding RESETPVC=YES on the X25.MCH statement causes PVCs to be reset when they are started.
- Coding RNRTIMER=*n* on the X25.MCH statement causes PVCs to be reset when an RNR packet has been received but has not been compensated by an RR packet.
- Coding RNRPKT=YES on the X25.MCH statement causes an RNR packet to be sent when a data packet is received and the buffer pool is depleted or the virtual route is blocked.
- Call setup commands from the GATE CTCP might not carry the X.25 flow control parameter values.
- You can use the DLCADDR keyword instead of, or in addition to, the DIALNO keyword on the PATH statement in the VTAM switched major node definition.

Migration from NPSI V3R7 to NPSI V3R8

Consider the following information when you migrate from NPSI V3R7 to V3R8:

NPSI appends a CV57 (DLC connection data control vector) containing the physical PU name to the REQCONT for a virtual circuit PU. This allows VTAM and NetView to correlate an X.25 virtual circuit to its MCH when the connection is established.

When migrating to SSP V4R4 or later and NCP V7R4 or later, you must code USERGEN=CBEX25 on the OPTIONS statement in the NCP generation.

Migration from NPSI V3R8 to X.25 ODLC

The following three keywords have been added to the X.25 generation statements to support the NCP X.25 ODLC function:

- ODLC=YESINO on the X.25 MCH statement, page 5-47
- DTETODTE=YESINO on the X.25 MCH statement, page 5-46
- CCITT={*value*} on the X.25 NET statement, page 5-80

Migration from NPSI V3R8 to V3R9

Users of NPSI V3R8 who use 3745 Communication Controllers can migrate to NPSI V3R9. The following section identifies new or modified keywords for consideration.

- X25.LINE Statement
 - RESETUNB
- X25.NET Statement
 - RESETUNB
- X25.MCH Statement
 - CUD0
 - FTPITH
 - FTPITHTO
 - GATEINFO
 - OWNER
 - RESETUNB
- X25.PU Statement
 - HPRQLIM
 - LLERP
- X25.VC Statement
 - HPRQLIM
 - LLERP
 - RESETUNB

Planning for the Configurator

The CF3720 configurator applies to the 3720 communication controller. The CF3745 configurator applies to the 3745 communication controller with or without the 3746 Model 900 attached.

The CF3720 and CF3745 configurators assist in planning your configuration by:

- Calculating the CCU utilization

You should limit the CCU utilization to a maximum of 90 percent to preserve response time and avoid slowdown due to buffer shortages.

- Calculating each physical circuit's utilization

IBM configurators compute the MCH load as a function of the traffic on the dependent virtual circuits. It is recommended that you limit the percentage of use of each leg (transmit and receive) to 60 percent to avoid excessive queueing on the MCH. Excessive queueing degrades response time and eventually causes slowdown due to buffer shortage.

The configurator assumes that the link level window is 7, which allows the configurator to assume full piggybacking at link level.

Note: Remember that link level piggybacking is achieved by coding the appropriate values for SPEED and T1TIMER on the X25.MCH statement.

The configurator takes into account the degree of piggybacking at the packet processing level by the receipt of what is coded for the number of input frames per link-level response. If the packet level processing window is less than 3, no piggybacking occurs. The coded response must be 1. If your packet level processing window is 3 or more, and if your inbound traffic is balanced by outbound traffic, full packet level processing piggybacking occurs. The coded response must be 7. The response to the number of output frames per link-level response must be 1. When answering for non-SNA DTEs, the following parameters apply:

- Code OUT PACING as 1.
- Code IN PACING as 0.
- Code VPACING so that it is equal to the value coded either at generation or in the switched major node (SMN).

Chapter 4. Defining X.25 Networks

Physical Circuits	4-2
Logical Channels	4-2
Virtual Circuits	4-5
PVC Definition	4-5
SVC Definition	4-6
SVCSC and Short Hold Mode Definitions	4-6
Call Definitions	4-8
Call-In Definitions	4-8
Call-Out Definitions	4-13
SNA Type 2.1 Node Definition	4-14
Associated Tables	4-15
SVCSC Considerations	4-17
Calling DTE Address Verification (X.25 ODLC and NPSI V3R5 and Later)	4-17
VTAM User-Exit (X.25 ODLC and NPSI V3R5 and Later)	4-18
Virtual Circuit to MCH Mapping (X.25 ODLC and NPSI V3R8 and Later)	4-18
Dynamic NETID (X.25 ODLC and NPSI V3R4 and Later)	4-18
Support of Performance and Accounting for NPM (X.25 ODLC and NPSI V3R4 and Later)	4-18
Accounting	4-19
Performance	4-20
Clear SVC on Inactivity Timeout	4-22
Enhanced X.21 Support	4-22
CTCP GATE Command Interface	4-23
REFCMS	4-23
LLC Selection	4-23
PVC Selection	4-24
SVC Selection	4-24
For Call-In	4-24
For Call-Out	4-26
LLC Type and Extended CTCP Selection	4-28
CTCP Selection Using the Address Extension Facility	4-29
System Generation Parameters	4-30
CTCP Definitions	4-30
DATE	4-30
GATE	4-31
PVC Specifications	4-31
SVC Specifications	4-31
Fast Connect Option	4-33
QOS (NPSI V3R4 and Later)	4-36
X25.MCH Statement	4-36
X25.LINE Statement	4-37
X25.VC Statement	4-37
PVC	4-38
SVC	4-38
Fast Transaction Processing Interface (NPSI V3R4 and Later)	4-39
PAD Support	4-40
Generation of PAD Parameters	4-41
PCNE-to-PCNE Considerations	4-41
Multichannel Link Compatibility	4-42

This chapter provides information needed to define your network configuration using NPSI, including:

- Physical circuits
- Logical channels
- Virtual circuits
- Logical link control selection
- CTCP definitions
- PAD support
- Generation of PAD parameters
- PCNE to PCNE considerations
- Multichannel link compatibility

Physical Circuits

A physical circuit, or multichannel link (MCH), is a hardware connection between two locations. NPSI defines physical circuits between the communication controller that contains NPSI and the X.25 connection. The X25.MCH statement is used to define a physical circuit and characteristics common to all the groups of logical channels associated with the physical circuit.

The following are examples of the physical-level interfaces specified in the X25.MCH statement:

- Disable timeout value
- Enable timeout value
- Activation after a VTAM cold start

The physical circuit must be defined to allow NPSI to operate the connection. The definition of the physical circuit includes the communication controller port address and a number of link-level parameters, such as:

- Frame size
- Frame window size
- Retry counts
- DTE or DCE distinction

Logical Channels

A *logical channel* is the path on which data travels either between its origin DTE and the PSDN, or between the PSDN and its destination DTE. The PSDN creates a virtual circuit by allowing communication between logical channels connected to two DTEs. When a virtual circuit is established, the PSDN and NPSI use a logical channel group number and logical channel number to identify the virtual circuit. These two numbers comprise the *logical channel identifier* for that virtual circuit. Figure 4-1 on page 4-3 is an example of a logical channel definition.

The logical channel identifier consists of three hexadecimal digits. Digit 1 is the logical channel group number; digits 2 and 3 are the logical channel number.

Each packet that NPSI exchanges with the PSDN contains, in its packet header, the logical channel identifier for the appropriate virtual circuit. NPSI uses the logical channel identifier to associate a packet with a particular virtual circuit and then sends the packet to the PSDN. The PSDN assigns a new logical channel identifier to route the packet through the network to its destination DTE. Also, the PSDN

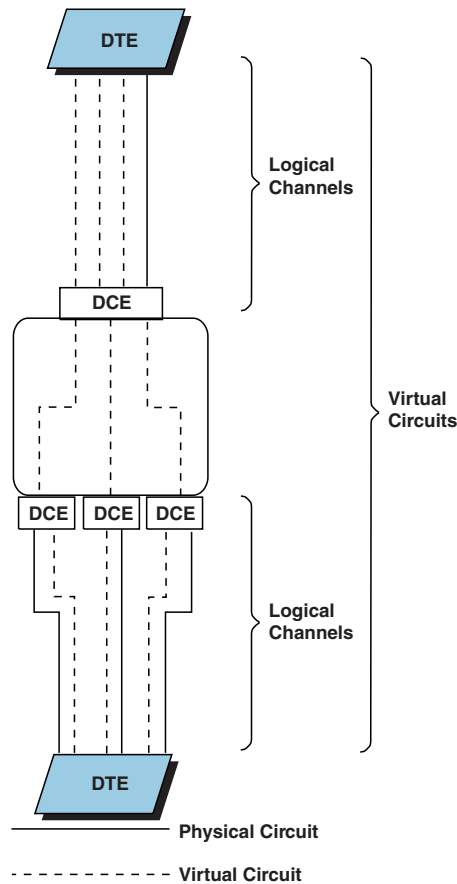


Figure 4-1. Logical Channel Definition

keeps a record of the pair of logical channel identifiers associated with each virtual circuit.

By keeping the logical channel pairings consistent, the PSDN ensures that data is not passed to an inappropriate destination. DTEs use the logical channel identifier from the PSDN to associate the logical channel with the virtual circuit.

Use the following statements and keywords to define logical channels to NPSI:

```
X25.MCH      LCGDEF=((1cg,1cnhi),(1cg,1cnhi),...)
X25.LCG      LCGN=number
X25.VC       LCN=(number1,number2)
X25.LINE     LCN=number
```

Several logical channels can be assigned to the same physical circuit.

The LCGDEF keyword of the X25.MCH statement specifies the logical channel groups associated with the defined physical circuit. The LCGDEF keyword also defines the greatest logical channel number within each logical channel group. This allows you to define a separate range of logical channel numbers for each logical channel group.

An X25.LCG statement follows each X25.MCH statement. One X25.LCG statement must be created for each logical channel group defined in the X25.MCH statement. The LCGN keyword of the X25.LCG statement defines the logical channel group

Defining X.25 Networks

number for all the virtual circuits specified by the subsequent X25.VC or X25.LINE statements.

Use the X25.VC statement or a combination of the X25.LINE, X25.PU, and possibly the X25.LU statements to define the virtual circuits that are members of the logical channel group defined by the X25.LCG statement. Code these statements following each X25.LCG statement. Use the LCN keyword of the X25.VC and X25.LINE statements to assign a logical channel number to each virtual circuit. The X25.VC and X25.LINE statements must be arranged in ascending order by logical channel number. The X25.LU statement defines logical units for the PVC DTEs that are associated with the virtual circuit for peripheral node connections.

Note: Additional X25.LINE and X25.PU statements not associated with a particular logical channel number are included to define resources to be used with SVCSC.

This coding creates a hierarchical relationship of the physical circuit to the logical channel group to the logical channel number. In addition, you must define each virtual circuit for one logical channel group before defining the next logical channel group. As a result, if a physical circuit contains logical channel groups 2 and 3, *all* of the definitions for logical channel group 2 *precede* the definitions for logical channel group 3. The following is an example of the order of these statements.

```
X25.LCG LCGN=2

X25.LINE LCN=1,...
X25.PU   ...
X25.LU   ...
X25.LU   ...
X25.VC   LCN=(2,5),...

X25.LCG LCGN=3

X25.LINE LCN=1,...
X25.PU   ...
X25.LU   ...
X25.VC   LCN=(2,5),...
```

Virtual Circuits

Virtual circuits are defined through the interaction of several statements. NPSI uses the interactions between these statements to generate NCP definitions. The virtual circuit definition includes designations for:

- PVC or SVC
- Logical link control (LLC)
- Call-in or call-out
- Packet window size
- Packet size

Virtual circuits differ from physical circuits in that virtual circuit connections are logical rather than physical. The use of virtual circuits provides a multiplexing technique that increases the use of the physical circuit. This path flexibility allows any DTE to connect with any other DTE that is connected to the PSDN.

The PSDN configuration determines which physical circuits are used.

PVC Definition

A PVC does not perform any dial procedures through the PSDN. PVCs simulate the action of a dedicated line and are activated with the physical circuit. Connection definitions are located in NPSI.

You can use the X25.VC statement to define PVCs that connect to non-SNA DTEs. The X25.VC statement produces one NCP LINE, PU, and LU statement for each virtual circuit.

Use the X25.LINE, X25.PU, and X25.LU statements to define PVCs that connect to SNA peripheral nodes. You can also use these statements to define PVCs that connect to non-SNA DTEs. Use the X25.LINE and X25.PU statements to define PVCs that connect to SNA subarea nodes.

You can use the X25.LINE, X25.PU, and X25.LU statements to create network names that meet a customized naming standard. Use statement labels as you do for NCP generation. You can also use the X25.VC statement with PRFLINE, PRFPU, PRFLU, SUFFIX, and HEXNAME to build customized names.

After the generation process is complete, your NPSI resources are represented as NCP LINE, PU, and possibly LU statements. These statements correspond to your original X25.LINE, X25.PU, and X25.LU statements, as appropriate, or to your original X25.VC statements. You can activate, deactivate, and test them as you would any other NCP resource.

Arrange the statements in ascending order by logical channel numbers within the same logical channel group, as shown in the sample program in Appendix B, "System Generation Examples."

SVC Definition

An SVC is established in response to either the arrival of an Incoming Call packet, or a request for an Incoming Call packet by a host SSCP. For an Incoming Call packet, the SVC is established when the Incoming Call packet originates as a Call Request packet sent by a remote DTE. For a request by a host SSCP, NPSI originates a Call Request packet that should be received as an Incoming Call packet by the remote DTE. An SVC is established either in response to the arrival of an Incoming Call packet originating as a Call Request packet sent by the remote DTE, or on request by a host SSCP that causes NPSI to originate a Call Request packet that should be received as an Incoming Call packet by the remote DTE.

To define SVCs, use either the X25.VC statement or the X25.LINE and X25.PU statement sequence.

Note: No X25.LU statement is associated with this definition.

The physical unit and, where appropriate, the logical unit definitions for an SVC are contained in members of the VTAMLST source library called the SMN. There can be one or more members within the VTAMLST library. SMNs contain definitions for all switched facilities that run under VTAM, whether or not they use NPSI. The NPSI switched resource entries contain the physical unit and, where appropriate, the logical unit definitions that include VTAM and NCP parameters used for resource activation.

The NCP LINE and PU statements that are generated either from the X25.LINE and X25.PU statements, or from the X25.VC statements appear in an order that is the reverse of the X25.LINE and X25.PU or X25.VC order of specification. This causes VTAM to select the LINE resource with the highest logical channel identifier when requesting an outgoing call. The network selects the lowest available logical channel identifier when presenting an Incoming Call packet, reducing the chance of a call collision. See Figure 7-1 on page 7-8 and Figure 7-2 on page 7-11.

The names defined for the PU in VTAM and NPSI *do not* need to match. The name used by VTAM and the NetView program is the name in the SMN. The PU defined in the NCP is used to create only the control block structure. To locate the proper PU statement in the SMN, use either the IDNUM/IDBLK combination or CPNAME.

SVCSC and Short Hold Mode Definitions

SVCSC provides connectivity between two subarea nodes over switched virtual circuits. Six keywords in the X25.MCH statement are associated with the SVCSC function; but only SVCINN is used by X.25 ODL:

SVCINN Is the number of simultaneously active switched subarea node connections. It represents the number of switched virtual circuits allocated to subarea node traffic. SVCINN must equal the number of LINE and PU statements defined with DSTNODE=INN,TYPE=S.

SVCINN=0 means that no SVCSC is on the MCH. The default value is 0.

SDRTCNT Is the number of retries attempted when the short hold mode (SHM) reconnection cannot be established during call setup and the switched XID exchange.

The default value is 0 (no retry).

SDRTIME Is the timer used between two retries.
The default value is 10 seconds.

SHM Specifies whether short hold mode is to be used by SVCSC on the MCH. SHM=NO specifies that no SVCSCs on the MCH use SHM. SHM=YES specifies that at least one SVCSC on the MCH uses SHM. The default is YES.

Short Hold Mode is used only if it is specified in the following four places:

- The X25.MCH statements of the two NPSIs that are in contact
- Both PATH definitions associated with the PU statements in the switched major nodes that are selected when the two link stations make contact

If SHM is not specified in all of these places, SHM is not used.

ACTIVTO Determines an inactivity timer that can be set for the MCH. The inactivity timer starts when the connection is no longer sending and receiving PIUs, and is reset when a PIU is scheduled to be sent or begins to be received.

The ACTIVTO keyword is coded on the X25.MCH statement. ACTIVTO can be used when operating with an Enterprise System/9370 processor through a telecommunication subsystem controller (TSC). The conditions under which the ACTIVTO mechanism is in effect is determined by the specification of the BRKCON keyword. ACTIVTO determines the maximum number of seconds of inactivity before NPSI clears the virtual circuit and sends an INOP message to the host. (The ES/9370 does not support SHM.)

An initial connection using the V NET,DIAL command must be repeated after an ACTIVTO.

See the description of SHM to determine when SHM can be in effect.

BRKCON The BRKCON keyword of the X25.MCH statement determines when the ACTIVTO timer mechanism comes into effect. Use the following values for the BRKCON keyword to determine when ACTIVTO is activated:

CONNECTO Specifies that the inactivity timer mechanism is always active.

NOWNERTO Specifies that the inactivity timer mechanism is active only when the link and link station are no longer owned by an SSCP.

NONE Specifies that the inactivity timer mechanism is never active.

The required virtual circuit definitions consist of two parts. The parts are the SNA and X.25 resource sets. This requirement is similar to the coding needed for the fast connect function.

Defining X.25 Networks

The SNA set of resources is defined with the X25.LINE and X25.PU statements. The X25.LINE statement must specify:

```
TYPE=S|SWITCHED  
DSTNODE=INN
```

The LCN keyword must be omitted in the X25.LINE statement. The X25.PU statement must specify PUTYPE=4, defining the definitions as applying to a subarea connection.

Note: The specification of PUTYPE=4 should be understood to include PU type 5, as the symbol to conform to style guidelines case with an ES/9370 processor through a TSC.

In addition, the X.25 resources must be defined using the X25.VC statement. The virtual circuits defined with this statement can be used for both subarea and peripheral node communication.

The connection of the SNA and X.25 resources is done during the establishment of the virtual circuit. For an incoming call, this connection is established *after* the pre-negotiation exchange because the call must be identified as a newly established connection or a short hold mode reconnection.

Note: When using the SVCSC function, note the following:

- The number of SNA resources must match the specification of the SVCINN keyword on the X25.MCH statement.
- The virtual circuits, defined for SVCSC, are taken into account when determining the default IDNUM *if* the LLCLIST keyword on the X25.MCH statement includes any of the following:
 - LLC0
 - LLC4
 - LLC5

Only single link transmission groups are supported for X.25.

For V3R4 and Later: Code TGN on the X25.PU statement.

Call Definitions

To perform a call-in or call-out procedure, X.25 must be operating with SVCs. PVCs do not perform any type of call operation. SVCs use a SMN in the VTAM statement library to complete the call operation.

See “SVCSC Considerations” on page 4-17 for more information about switched subarea node connections.

Call-In Definitions

The PUTYPE, CPNAME, IDBLK, IDNUM, and IDBLKLCG (NPSI V3R8 or Later) keywords are used for the call-in definitions.

PUTYPE: The PUTYPE keyword for the PU statement in the SMN specifies the physical unit type. Code PUTYPE=1 for LLC types 0, 4, and 5. Code the PUTYPE keyword according to the type of device for LLC types 2 and 3. You can obtain the PUTYPE of an SNA peripheral node from device documentation. You must define an SNA type 2.1 node as a PU type 2.

CPNAME: For PU type 2.1 node switched resources, the identification is obtained from the resource, but two possibilities exist. The link-level identification procedure can be the same as for other SNA switched resources, or it can result in obtaining a CPNAME value rather than the IDNUM and IDBLK values. When a CPNAME value is obtained, the CPNAME value must match the CPNAME keyword of a PU statement in an active SMN.

The specification of IDNUM and CPNAME is usually part of the product customization for SNA switched resources.

SMN Identification: When a switched resource either contacts or is contacted by NCP, NCP presents a resource identification to the host (VTAM). The type of identification presented depends on the nature of the link-level identification exchange procedure.

NPSI allows you to define the IDBLK value using the X25.MCH statement during system generation and the IDNUM values using the X25.LINE and X25.VC statements.

When an Incoming Call packet specifies a non-SNA LLC, NPSI uses the IDBLK and IDNUM defined values or their default values described in the preceding paragraph to create the identification number for selecting the correct SMN entry.

When an Incoming Call packet specifies an SNA LLC, IDNUM and IDBLK are specified within the XID sent by the remote SNA device.

IDBLK: You can define the IDBLK value during system generation using the IDBLKP, IDBLKC, and IDBLKG keywords of the X25.MCH statement. The value generated by the IDBLK parameters has a range of 13 numbers (X'62' to X'6E'). The value that you define for IDBLK allows for PAD, PCNE, and GATE connections to coexist on the same MCH.

Keyword	Description
IDBLKC	Specifies a PCNE connection
IDBLKP	Specifies a PAD connection
IDBLKG	Specifies a GATE connection

If you do not specify the IDBLK value, the default value is X'003', but because this value could already be used by another SNA resource within the network, it is not recommended.

In the case of SNA switched resources, the identification is obtained from the resource. The value of IDBLK is fixed according to the type of product; for example, X'017' is the value for the 3174 Establishment Controller. The required value is contained in the product documentation.

IDBLKLCG (NPSI V3R8 or Later): You can establish relationships between logical channel group numbers (LCGNs) and IDBLKs to allow different applications to be invoked according to the X.25 virtual circuit number. There are 13 ID blocks (062 to 06E) dedicated to NPSI; this gives a possibility of 13 different connections to various applications. This function applies for non-SNA LLCs only (PCNE, GATE or PAD).

Defining X.25 Networks

To establish the relationships between the X.25 LCGNs and the IDBLKs for both incoming and outgoing calls, at system generation time, code the IDBLKLCG=lcg,idblock parameter on the X25.MCH statement. For example, if you code:

```
IDBLKLCG=((1,062),(2,064))
```

The 062 will be the IDBLK number inserted in the XID for calls established over virtual circuits owned by logical channel group (LCG) number 1. The 064 will be the IDBLK number inserted in the XID for calls established over virtual circuits owned by logical channel group (LCG) number 2.

When you code VTAM so that the LUs belonging to the PUs with IDBLK=062 are logged on to application A and the LUs belonging to the PUs with IDBLK=064 are logged on to application B, then:

- A terminal connected through all virtual circuits owned by LCG number 1 would be in session with application A
- A terminal connected through all virtual circuits owned by LCG number 2 would be in session with application B.

IDNUM: In the case of SNA switched resources, the identification is obtained from the resource.

In the case of non-SNA LLC type 0 without subaddressing, the IDNUM is obtained from the Incoming Call packet user data field (CUD=C0zzzzz..., where C0 is the protocol identifier for LLC type 0 and zzzzz is the IDNUM). See Table 4-1 on page 4-12 for exceptions.

In the case of other non-SNA resources supported by NPSI, the first digit of the IDNUM value is obtained from the value of the X25.IDNUMH keyword on the BUILD definition statement. The last 4 digits of the IDNUM value are obtained from the value of the IDNUMT keyword on either the X25.LINE or X25.VC statements. The identification must match the IDNUM and IDBLK operands of a PU statement in an active SMN.

Defining the IDNUMT keyword restricts NPSI from generating any default values for other SVCs. Therefore, during system generation, you must specify either all of the IDNUMT values or none of the IDNUMT values. NPSI stops the generation process if you define only some, but not all, of the IDNUMT values for non-SNA SVCs.

When you define the SVCs on the X25.VC statement during system generation, NPSI automatically increments the IDNUMT value by 2. Therefore, if the X25.VC statement defines 10 SVCs, X25.IDNUMH=2 is coded on the BUILD statement, and the IDNUMT value is X'1000', the default IDNUM values for each SVC will be:

- X'21000'
- X'21002'
- X'21004'
- X'21006'
- X'21008'
- X'2100A'
- X'2100C'
- X'2100E'
- X'21010'
- X'21012'

The IDNUMT values must be even.

If IDNUMT is not coded for non-SNA switched resources, NPSI generates default IDNUM values. This creation is based on the sequence of the virtual circuit definition and the specification of the X25.IDNUMH keyword of the BUILD statement. The identification must match the IDNUM and IDBLK operands of a PU statement in an active SMN. For example, if X25.IDNUMH=2 is coded on the BUILD statement and IDNUMT is not coded for non-SNA switched resources, NPSI generates the following IDNUM values for 10 SVCs:

- X'20002'
- X'20004'
- X'20006'
- X'20008'
- X'2000A'
- X'2000C'
- X'2000F'
- X'20010'
- X'20012'
- X'20014'

Notes:

1. Default IDNUM values start at X'0002' and increment by two. NPSI generates IDNUM default values in *reverse* order from the way the SVCs were initially defined.
2. At generation, NPSI generates a default IDNUM for each SVC, except in the following cases:
 - SVCs on a fast connect MCH
 - SVCs on an MCH where LLCLIST does not require LLC0, LLC4, or LLC5
 - SVCs on an FTPI MCH

These identification numbers appear in the order in which the SVCs appear in the Resource Resolution Table. This order corresponds with the resource order in the VTAMLST NCP source statements, which is the *reverse* order of the NPSI sources statements.

Defining X.25 Networks

Table 4-1. IDNUM Assignment

LLC Type	Physical Circuit	Call-In	Call-Out
LLC type 0	GATE=NO	CUD field	ZZZZZ field of DIALNO or DLCADDR (see note)
	GATE=DEDICAT	DEFAULT IDNUM or user-specified (even numbers)	DEFAULT IDNUM or user-specified (even numbers)
	GATE=GENERAL SUBADDR=NO	CUD field	ZZZZZ field of DIALNO or DLCADDR, unless null CUD selects LLC0 (see note)
	GATE=GENERAL SUBADDR=YES	DEFAULT IDNUM or user-specified (even numbers)	ZZZZZ field of DIALNO or DLCADDR (see note)
LLC type 2		PU ID from remote SNA DTE	PU ID from remote SNA DTE
LLC type 3 peripheral node		PU ID from remote SNA DTE	PU ID from remote SNA DTE
LLC type 3 subarea node		Value specified in IDNUM of remote end PATH statement	Value specified in IDNUM of remote end PATH statement
LLC type 4		DEFAULT IDNUM or user-specified (even numbers)	DEFAULT IDNUM or user-specified (even numbers)
LLC type 5	GATE=NO or GATE=GENERAL	DEFAULT IDNUM or user-specified (even numbers)	ZZZZZ field of DIALNO or DLCADDR (see note)
	GATE=DEDICAT	DEFAULT IDNUM or user-specified (even numbers)	DEFAULT IDNUM or user-specified (even numbers)

X.25 ODLC and NPSI V3R7 and later: You can code DLCADDR instead of or with DIALNO on the PATH statement in a VTAM switched major node.

Note: The IDNUM creation process results in a limit of 32 768 uniquely numbered values. The maximum number of SVCs using a default IDNUM that can be created in one NCP/NPSI load module is 32 768. The IDNUM creation process allows 16 distinct values for the X25.IDNUMH keyword. This means that no more than 16 communication controllers can be under the control of one host running NPSI in contact with non-SNA resources. (It is common usage to relate an IDNUMH to a communication controller.)

Figure 4-2 on page 4-13 shows definition examples of the various IDBLK values.

No separation between LLC's:

```
NCP1:   IDBLKG=062   IDNUMH=0   (32000 SVC's)
        IDBLKP=062
        IDBLKC=062
```

With separation for GATE/PAD/PCNE:

```
NCP2:   IDBLKG=062   IDNUMH=1   (32000 SVC's)
        IDBLKP=063
        IDBLKC=064
```

With separation of GATE from PAD/PCNE:

```
NCP3:   IDBLKG=062   IDNUMH=2   (32000 SVC's)
        IDBLKP=063
        IDBLKC=063
```

With separation for GATE/PAD/PCNE:

```
NCP16:  IDBLKG=062   IDNUMH=F   (32000 SVC's)
        IDBLKP=063
        IDBLKC=064
```

With separation for GATE/PAD/PCNE:

```
NCP17:  IDBLKG=065   IDNUMH=0   (32000 SVC's)
        IDBLKP=066
        IDBLKC=067
```

Figure 4-2. Examples of Various IDBLK Values

NCP Anonymous Caller ID Definitions X.25 supports the NCP anonymous caller function. However, you must use the ID keyword on the X25.LINE and X25.VC statements. This applies to MCHs supporting only one kind of SVC (either peripheral SNA or PCNE SVCs without subaddressing or DATE). You must define the ID keyword on all SVCs for a given MCH.

Call-Out Definitions

The PUTYPE, IDBLK, and IDNUM keywords are used for the call-out definitions. The PATH definition statement is also used for the call-out definitions.

PUTYPE: The PUTYPE keyword for the PU statement in the SMN specifies the physical unit type. Code PUTYPE=1 for LLC types 0, 4, and 5. Code the PUTYPE keyword according to the NCP rules for LLC types 2 and 3. See the *NCP, SSP, and EP Resource Definition Guide* for NCP rules.

IDBLK and IDNUM: For SNA remote DTEs, the parameters specified for these keywords must match the IDBLK and the IDNUM of the remote SNA DTE.

For non-SNA remote DTEs, the IDBLK defaults to X'003' (for migration purposes), and the IDNUM must match the value specified in the ZZZZZ field of the DIALNO or DLCADDR keywords.

Rather than using X'003', which is the IDBLK default value for migration purposes only, it is recommended that you specify IDBLK by coding the IDBLKC, IDBLKG, or IDBLKP keywords on the X25.MCH statement using the range X'62'–X'6E'.

PATH Definition Statement: The PATH definition statement in a switched major node is used to define a dial-out path to a remote PU. Up to 256 PATH statements can be specified for each PU. The PATH statement must immediately follow the PU statement that defines the associated PU. VTAM searches the PATH statements for an available path in the order specified in the configuration deck.

The ZZZZZ field of the DIALNO or DLCADDR keywords must be coded for LLC type 0 and LLC type 5, and must match the IDNUM specified in the PU statement corresponding to the involved PATH statement.

For LLC type 0, NPSI builds the CUD field containing the following value:

```
X'C0kkkkk0'
```

where kkkkk=ZZZZZ + 1.

See Chapter 6, “Coding NCP and VTAM Parameters for X.25 Support,” for more detailed information on the required keywords and their exact coding requirements.

SNA Type 2.1 Node Definition

SNA type 2.1 nodes can communicate with each other, as well as with applications in VTAM. VTAM determines whether a peripheral node is PU type 2.1 or PU type 2.0 either during activation of the node or at contact time for switched nodes.

In a switched environment, VTAM can locate the appropriate PU/LU pair either by using the normal IDNUM/IDBLK combination or by using the CPNAME keyword on the PU statement in a switched major node source statement member of VTAM.

For call-in and call-out definitions, SNA type 2.1 uses QLLC with X'C3' in CUD0. For a call-out, LLC type 3 is used as in the normal peripheral node QLLC.

XID=YES must be coded, and PUTYPE=2 should be specified, in the X25.PU statement for PVCs of SNA type 2.1 nodes.

Enhanced SNA Type 2.1 Boundary Function Support—Casual Connection:

The enhanced SNA type 2.1 boundary function support provided in NPSI V3R3 and later releases extends the NCP's X.25 boundary node support, by allowing a primary SNA type 2.1 peripheral node to be attached to an NCP through an NPSI that is acting as the secondary partner.

The enhanced SNA type 2.1 boundary function support is also called *casual connection*. Casual connection allows a VTAM, NCP, and NPSI configuration, working as a node T2.1, to communicate with another node T2.1 over a peripheral link. On a subarea link, you must define the remote partner to the local VTAM. On a peripheral link, VTAM does not need to know about the remote partner. The

remote partner is an SNA type 2.1 node working as a primary or secondary mode. This type 2.1 node can be a VTAM, NCP, NPSI configuration. This type of configuration is called a composite end node (CEN). For a casual connection, only one CEN needs to support the casual connection function to add role negotiation and secondary support. The other CEN should be LEN-level, but it does not need to support casual connection.

The following keywords apply to casual connection:

- CCXDELAY on the X25.MCH statement for SVC
- MODE=SEC on the X25.LINE statement for PVC
- ROLE=PRIINEG on the X25.MCH statement for SVC

Associated Tables

In addition to defining LLC types, you must define parameters that are to be associated with each virtual circuit. The following are two types of virtual circuit parameters:

- Virtual circuit connection variables

The virtual circuit connection variables apply to all virtual circuits and include:

- Maximum packet size (in the absence of packet size negotiation)
- Packet-level window size (in the absence of packet window size negotiation)
- Parameters to prevent NCP slowdown

- Optional user facilities

The optional user facilities apply to X.25 facility codes and user data copied into the Call Request packet when NPSI makes an outgoing call for LLC types where a CTCP does not create the Call Request packet directly.

X25.VCCPT Statement: Virtual circuit connection variables are defined using the X25.VCCPT statements. These statements create a table that NPSI uses to specify the virtual circuit connection parameters. Each X25.VCCPT statement is one entry in this table. See “X25.VCCPT Statement” on page 5-111 for the format of, and coding information about, X25.VCCPT.

Call-In Specifications: For call-in, to define the VCCPT entry, use the VCCINDX keyword on either the X25.VC or X25.LINE statement. This keyword corresponds to the INDEX keyword of the X25.VCCPT statement. For example, if you want to use the entry specified by INDEX=2 on the X25.VCCPT statement, code VCCINDX=2 on the statement that you are using to define your virtual circuit. When you use a GATE or DATE CTCP, the packet size and packet window size are selected through the CALL ACCEPTED command. Byte 3 of this command contains the packet window size. Bytes 4 and 5 contain the packet size. All sizes are in hexadecimal.

Values for packet size and packet window size can be specified in the Flow Control Parameter Negotiation facility in the Incoming Call packet. This allows an SVC to work with a different set of values from those defined by the VCCINDX keyword. See Appendix A, “Defining X.25 PSDN Facilities,” for more information.

Defining X.25 Networks

NPSI slowdown parameter definitions for virtual circuits are not affected by the use of GATE or DATE. These values are taken from the applicable VCCPT entry pointed to by the VCCINDX keyword on the virtual circuit definition.

Call-Out Specifications: For call-out, to define the VCCPT entry, use the XX component of the DIALNO or DLCADDR keywords on the VTAM PATH statement. The XX component value specifies which VCCPT entry to use, or designates the entry defined on the X25.VC or the X25.LINE statement. Thus, the XX field can be used to override the default VCCPT for the virtual circuit. See “Activating Calls that Use SVCSC” on page 7-4 for more information.

Specify the XX as zero (“00”) to designate the use of the VCCINDX on the X25.VC or X25.LINE statement. When you use a GATE or DATE CTCP, the packet size and packet window size are selected through the CALL REQUEST command from the CTCP to GATE or DATE. Byte 3 of this command contains the packet window size; bytes 4 and 5 contain the packet size. All sizes are in hexadecimal.

V3R7 and Later: If the GATE CTCP does not set the packet size and window size in the call setup command, NPSI gets these values from Call Setup packets or from the VCCPT definitions.

When you specify INTFAC=YES on the X25.MCH statement, NPSI will interpret and use the flow control parameters that are contained in the Call Connected packet under GATE and DATE. Therefore, NPSI will clear fewer calls because of incompatible packet sizes and window sizes.

NPSI slowdown parameter definitions for virtual circuits are not affected by the use of GATE or DATE. Their values are taken from the applicable VCCPT entry pointed to by the VCCINDX keyword on the virtual circuit definition.

X25.OUFT Statement: Use the X25.OUFT statements to create a table of optional user facility fields and a call user data (CUD) field for Call Request packets. NPSI uses these fields to specify the optional user facilities used on the virtual circuits. Each X25.OUFT statement defines one table entry. See “X25.OUFT Statement” on page 5-87 for more information on the X25.OUFT statement coding.

X25.OUFT applies only to a call-out from NPSI, because the optional user facilities field and the CUD field created are placed in the Call Request packet.

Call-Out Specifications: When performing a call-out, NPSI obtains the OUFT entry through the use of the YY component of the DIALNO or DLCADDR keywords on the VTAM PATH statement. This component specifies the OUFT entry or specifies using the OUFT entry defined on either the X25.VC or the X25.LINE statement. A X'00' value indicates that the default value coded on the X25.VC or X25.LINE statement should be used. Any non-zero value specifies an OUFT table entry. You can use the DIALNO keyword to override the default OUFT entry.

When you use GATE or DATE, the CTCP creates the optional user facilities and CUD field. The CTCP sends them in the Call Request packet at the end of the CALL REQUEST command sent to NPSI.

SVCSC Considerations

The PU statement involved in the SVCSC connection is located by VTAM using the following three keywords:

- NETID
- SUBAREA
- IDNUM

The PATH and the PU to be used with the connection is determined if and when these three parameters match the values obtained from the remote subarea station in the exchange of identification process.

The following mechanisms, which are based on the integrity of the network, are considered when using SVCSC within switched major nodes.

- Calling DTE address

X.25 networks always provide the address of the calling DTE. The called DTE can then check this address against a list of authorized callers. This mechanism, which uses the VERIFY keyword of the PATH statement, is optional.

VTAM does not support the outbound verification of the DTE address. As a result, when used with NPSI, the VERIFY keyword on the PATH statement must be coded as VERIFY=IN.

- Called DTE address

Call Redirection and Hunt Group subscription options are supported except when SHM is used on the virtual circuit. If SHM is used in either of these options, the SHM reconnection may fail.

Call Redirection is a PSDN subscription option that allows incoming calls, destined for a DTE that is out of order, to be redirected. When a call is redirected, a search for a specified group of DTEs (Hunt Group) is initiated.

- CUG

The CUG facilities enable the PSDN to control access to or from DTEs that belong to the CUG. CUG facilities can be subscribed to so that a Call Request is only accepted by the PSDN when the called DTE belongs to the same CUG as the calling DTE. This situation is particularly relevant to SVCSC.

Calling DTE Address Verification (X.25 ODLC and NPSI V3R5 and Later)

X.25 ODLC and NPSI always append a CV25 to the REQCONT sent to VTAM. This CV25 contains the calling DTE address and the CUD from the incoming call packet. The CV25 is checked in the host by VTAM if VERIFY/VERID was coded (BNN QLLC/PSH), or by a configuration services installation user exit, if one is coded (non-SNA LLCs). This exit is not useful in the case of GATE/DATE because the CTCP receives the call packet and can perform all the necessary checking.

Once IDBLK and IDNUM identify the SMN PU, VTAM performs an additional verification if VERIFY=IN and VERID=(addr1,addr2,...) has been coded on the PATH statement of this PU. VTAM compares the calling DTE address to the list of authorized addresses coded on the VERID statement. If no corresponding address is found, the connection is denied.

VTAM User-Exit (X.25 ODLC and NPSI V3R5 and Later)

VTAM allows you to code an exit routine that is scheduled whenever a call is received from a terminal that is not identified in the SMN. The user exit associates a calling DTE address (set by X.25 in a CV25 appended to the REQCONT) to a unique PU name, and dynamically creates an entry in the SMN. See *X.25 NPSI Version 3 Host Programming* for more information about coding the VTAM user-exit.

Virtual Circuit to MCH Mapping (X.25 ODLC and NPSI V3R8 and Later)

X.25 ODLC and NPSI build and append a CV57 (DLC Connection Data Control Vector) containing the physical line name to the REQCONT for a virtual circuit PU. This allows VTAM and NetView to correlate X.25 virtual circuits to their physical line (X.25 MCH) at connection establishment time.

Dynamic NETID (X.25 ODLC and NPSI V3R4 and Later)

The dynamic NETID enhancement allows for the dynamic addition and deletion of NETIDs (networks) without operator intervention. This allows NETID=(name,ANY) to be specified on the GROUP statement for leased lines, in the same way that it is currently specified on switched subarea lines. The dynamic NETID enhancement allows you to specify the number of times model network replication can occur.

When an undefined NETID is received, NCP uses one of the available dummy networks created during replication of the model network. The NCP marks the NETID as dynamic. When all users of this dynamic network are terminated, the network is marked as a dummy network and can be used again.

With NCP NETID processing support, the NETID=(name,ANY) keyword for switched subarea groups is allowed on leased subarea GROUPs. If you do not specify NETID=(name,ANY), the link station activation varies only slightly, and the NETID cannot change.

When you specify NETID=(name,ANY), NCP chooses a NETID during the XID exchange, unless a NETID is received. If a NETID is not received, NCP uses the name specified as the default, or native network ID.

Support of Performance and Accounting for NPM (X.25 ODLC and NPSI V3R4 and Later)

X.25 ODLC and NPSI support the interface to the NetView Performance Monitor (NPM) for both accounting and performance data. The NPM support of the NCP is used with the NPM component to forward either performance data, accounting data, or both on SNA resources. SNA is structured to send this type of information through the line and PU.

As a result of this structure, data is obtained through the MCH line (link layer), MCH PU (packet layer), and VC PU (information about a specific host connection). You cannot obtain information by virtual circuit. Information is available only on connections to the host. NPM gives you a representation from the point of view of the host.

Accounting

X.25 ODLC and NPSI report accounting information through the NPM. You must specify that you want accounting information by using the following keywords on the BUILD statement:

- SESSACC
- X25.BYTTHROD
- X25.SEGTHROD

You can restrict accounting to specific LLCs on specified lines by listing the desired LLCs on the ACCOUNT keyword on the X25.MCH statement. ACCOUNT=NO specifies that accounting data is not supported.

The accounting function requires the specification of several keywords and operands. You should thoroughly read this section before you enable accounting. You must do the following steps to initiate the accounting function:

- BUILD statement
 - Ensure that NPA=YES is specified.
 - Add SESSACC=(YES,ALL).
 - Add X25.BYTTHROD and X25.SEGTHROD keywords.

Note: The X25.BYTTHROD keyword defines the size of the byte threshold. Exceeding this threshold causes NPSI (NCP) to forward performance data to the host. The X25.SEGTHROD keyword defines the size of the segment threshold. Exceeding this threshold causes NPSI (NCP) to forward performance data to the host. Each segment has a fixed length of 64 bytes.

You should not code the X25.BYTTHROD and X25.SEGTHROD keywords if the ACCOUNT keyword is not specified on any of the X25.MCH statements.

- X25.MCH statement
 - Specify the ACCOUNT keyword as either YES or *llclist*.
 - Specify the NPADTEAD keyword to define the DTE address for the X25.MCH.
- GENEND
 - UACCTNG is defaulted to the correct module name if there is at least one ACCOUNT keyword specified on the X25.MCH statement.

If you are enabling and disabling accounting, you should ensure that all appropriate keywords and operands are correctly specified.

Table 4-2 on page 4-20 shows the statistics that are created by the NPM accounting function. You can use this data to provide a cost analysis, charge-back system, or other statistical accounting information reports.

See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information about specific fields and their respective formats.

Table 4-2. NPM Accounting Fields

Bilateral CUG selection facility
Called complementary address (from CUD)
Called DTE address
Calling DTE address
Called DTE address extension
Calling DTE address extension
Intermediate gateway address
Logical channel number
Reverse charging facility
Sequence number
Switched resource identifier
Throughput classes
Time stamp
Total number of segments inbound
Total number of segments outbound
Total number of bytes sent
Total number of bytes received
Total number of Reset packets (DTE originate)
Total number of Interrupt packets (DTE originate)
Transit delay
Type of bill (private or PSDN)
Type of virtual circuit
Window sizes (in : out)

Performance

X.25 ODLC and NPSI can obtain information about performance characteristics of the SNA connections through NPM. Performance data is available on the MCH line, MCH PU, and virtual circuit PU. As a result, you can obtain information at the link layer, packet layer, and specific to a connection.

The following are the system generation keywords that must be specified to activate the performance monitor:

- NPACOLL

The NPACOLL keyword is coded on the X25.MCH statement and specifies that performance data can be collected at three levels: MCH line, MCH PU, and virtual circuit PU.

Note: MCHLINE, MCHPU, and VCPU can be coded on the NPACOLL keyword if NPA=YES is coded on the BUILD statement.

- NPPVCN

Notes:

1. You must specify the NPPVCN keyword on the NPSI X25.MCH statement to define the number of virtual resources to be monitored concurrently.
2. The NPPVCN keyword is required if NPACOLL=VCPU is coded on an NPSI X25.MCH statement.
3. The NPPVCN keyword is not used for NCP X.25 ODL.

These specifications define the amount of space to be reserved in NPSI for obtaining performance information.

Note: The NPPVCN keyword specifies concurrent monitoring. If you have more than the number of virtual circuits eligible for monitoring, but the virtual circuits are not active at the same time, you can reduce the number specified on NPPVCN.

Table 4-3 shows the number of bytes allocated for NPSI performance processing. You can calculate the number of bytes required for performance monitoring.

<i>Table 4-3. Number of Bytes Required for NPSI Performance Monitoring</i>	
Type of Resource	Number of Bytes
MCH LINE	92
MCH PU	116
Virtual Circuit PU	100*nn where: nn is the number of virtual resources (switched and leased) needed for performance data

Tables 4-4 through 4-6 show the reported fields for each level. This information can be used either by a program created by you, or by an IBM-supplied product, such as the service level reporter (SLR).

See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information about the specific fields and their associated formats.

<i>Table 4-4. NPM Fields for MCH Link (Link Level)</i>	
Total I-frames (packets) sent	
Total I-frames (packets) received	
Total bytes sent (data and control)	
Total bytes received (data and control)	
Total RR frames sent	
Total RR frames received	
Total RNR frames and reject frames sent	
Total RNR frames and reject frames received	
Retransmitted frames count	
Retransmitted byte count	
Internal queue length	

Defining X.25 Networks

Table 4-5. NPM Fields for MCH PU (Packet Level)

Total number of outbound connections (Call Request packets)
Total number of inbound connections (Incoming Call packets)
Total number of outbound disconnections (Clear Request packets)
Total number of inbound disconnections (Clear Indication packets)
Current number of virtual circuits established
Number of new virtual circuits established during this interval
Total number of Data packets sent
Total number of Data packets received
Total number of RNR packets sent
Total number of RNR packets received
Total number of SVCSC SHM reconnections

Table 4-6. NPM Fields for Virtual Circuit PU (Virtual Circuit Level)

Total number of Data packets sent
Total number of Data packets received
Total number of bytes sent
Total number of bytes received
Total number of Data packets sent with the M-bit
Total number of Data packets received with the M-bit
Total number of Data packets sent with the D-bit
Total number of Data packets received with the D-bit

Note: You should use a naming convention for the virtual circuit LINEs and PUs on which performance data is collected because the NPM panel displays this PU name and not the PU name given in the switched major node.

Clear SVC on Inactivity Timeout

NPSI sets a timer based on the INACTO parameter. If there is no activity before the NPSI timer expires, NPSI clears and frees the SVC connection. Monitoring does not occur on subarea, GATE, or DATE virtual circuits.

Enhanced X.21 Support

Enhanced X.21 support allows CTCP to establish more than one virtual circuit session on an MCH dedicated to X.21 switched connections. Enhanced X.21 support provides X.21 Call collision security, which prevents the establishment of a virtual circuit session with an unidentified remote party through a CALL REQUEST command code. See *X.25 NPSI Version 3 Host Programming* for more information about Enhanced X.21 support.

CTCP GATE Command Interface

A new CALL REQUEST command (X'8B') is added to the already existing commands of the CTCP GATE interface.

In V3R4 and later releases, to establish the X.21 ISDN switched connection before the first virtual circuit session establishment, the CTCP sends the X'8B' CALL REQUEST command to GATE.

To establish a virtual circuit session on an already established X.21 ISDN switched connection, the CTCP sends the current X'0B' CALL REQUEST command to GATE.

RECFMS

The alert RECFMS 77 with MKBMDREC=X'9A' issued by NPSI on an X.21 switched connection failure is completed by the last call progress signal value returned by the network. This call progress signal value is set in the last byte of Vector 3 of the RECFMS. For the alert RECFMS, Qualifier 3 has the following format:

QUALIFIER 3 : 9A0000cc

Note: The last CPS value returned by the network is cc.

LLC Selection

NPSI connects through five LLC types as described in “Logical Link Control (LLC) Types” on page 3-26. LLCs define the DTE type to which the virtual circuit connects. DTE types include SNA peripheral nodes, subarea connections, and non-SNA DTEs. Each DTE type requires different data flow processing. X.25 ODLC supports only LLC type 3.

In addition, NPSI allows you to use a CTCP to control the virtual circuit. This support is supplied through either the GATE or the DATE function of NPSI.

For more information on GATE and DATE see “GATE” on page 4-31 and “DATE” on page 4-30. For detailed programming information, see *X.25 NPSI Version 3 Host Programming*.

When NPSI uses GATE or DATE, NPSI does not process X.25 commands. Instead, NPSI places the command in a buffer and forwards it to the CTCP. The CTCP initiates the response to the command.

NPSI processes a virtual circuit based on whether the virtual circuit is a PVC or an SVC, and whether a CTCP will be used.

PVC Selection

To define the LLC for a PVC, use the LLC keyword of the X25.VC or X25.LINE statement. The LLC keyword specifies the LLC type of the virtual circuit. The LLC for the PVC always stays the same.

Using a CTCP with a PVC gives you better session control, because you can use the CTCP to monitor use and to provide more security.

With GATE, code the OSITYPE keyword of the X25.VC or the X25.LINE statement to specify the CTCP.

SVC Selection

SVC definitions require careful planning, because SVCs are selected dynamically when calls are established. An SVC can be defined as any LLC type.

For Call-In

For an SVC used for call-in, define the LLC type in one of the following:

- Byte 0 of the CALL USER DATA field (CUD0)
- The subaddressing field of the incoming Call Request packet
- The CTCP response to an Incoming Call packet (DATE only)

NCP X.25 ODLC does LLC selection using CUD0, but only LLC3 can be selected (BNN and INN).

Using CUD0 for LLC Type and CTCP Selection: Table 4-7 shows default CUD0 values and the associated LLC types.

Table 4-7. CUD0 Values and Associated LLC Types

CUD0	LLC Type
X'01'	Type 5 (PAD)
X'41'	Type 5 (PAD)
X'51'	Type 5 (PAD)
X'81'	Type 5 (PAD)
X'C0'	Type 0 (PCNE)
X'CC'	Type 0 (PCNE)
X'C2'	Type 2 (PSH)
X'C3'	Type 3 (Peripheral node QLLC)
X'C4'	Type 4 (GATE)
X'CA'	Type 2 (PSH) V3R6 and later
X'CB'	Type 3 (BNN QLLC) V3R6 and later
X'E3'	Type 3 (Subarea node QLLC)
X'EB'	Type 3 (INN QLLC) V3R6 and later

You can also use the CUD0 field to specify the CTCP to use for GATE processing. See "GATE" on page 4-31 for more information about GATE processing.

Using Subaddressing for LLC Type Selection: When you define subaddressing to select LLC type, NPSI uses the *last* digit of the called DTE address to select the LLC type. NPSI assumes the last digit to be a subaddress digit. CCITT Recommendation X.121 defines the format of the addresses used by Recommendation X.25 and also specifies a maximum address length of 14 digits. NPSI allows up to

15 digits; thus, one address digit is always available. The network ignores address digits after those actually needed for addressing.

Note: NPSI cannot tell whether the last address digit is part of the address or is added by the calling DTE for LLC selection purposes.

The following example shows how you can define subaddressing digits to select LLC types:

```
LLC0=(1,2),LLC4=(0,3,4,7),LLC5=(6,5),LLCI=(8)
```

The coding shown above produces the following correspondence:

- Type 0 (PCNE) if 1 or 2 is the last subaddressing digit.
- Type 4 (GATE) if 0, 3, 4, or 7 is the last subaddressing digit.
- Type 5 (PAD) if 5 or 6 is the last subaddressing digit.
- Type 3 (Subarea node QLLC) if 8 is the last subaddressing digit.

When LLC type 4 is selected through subaddressing, if not working in fast connect, NPSI will always connect the virtual circuit to the first CTCP. Even though a GATE CTCP is not active and is not intended to become active, to use subaddressing you must code both GATE=GENERAL and SUBADDR=YES on the X25.MCH statement.

Note: If LLC type selection through subaddressing is specified, but the last digit of the called DTE address in the Incoming Call packet does not correspond to one of the generation-defined subaddress digits, LLC type and CTCP selection is performed using CUD0.

Using DATE for LLC Type Selection: An SVC using DATE is defined with the GATE=DEDICAT keyword of the X25.MCH statement. Such an SVC can be a type 0, 2, 3, or 5 virtual circuit. Type selection occurs when the incoming Call Request packet is passed to the DATE CTCP. The CTCP reads information in the CALL REQUEST command, selects the type, and sends a CALL ACCEPTED command to the DATE portion of NPSI. Byte 6 of this command specifies the virtual circuit type.

Using Subaddressing for CTCP Selection (NPSI V3R4 and Later): When you define subaddressing for CTCP, you must use either the FC option or the FTPI by coding either CONNECT=SUBD or FTPI=SUBD on the X25.MCH statement.

When you define subaddressing to select a CTCP, NPSI uses the last digit, or the last 2 digits, of the called DTE address to select the CTCP. NPSI will use 2 digits to select a CTCP if the SUBD values on the X25.MCH statement have 2 digits.

CCITT Recommendation X.21 defines the format of the addresses used by Recommendation X.25 and specifies a maximum address length of 14 digits. X.25 NPSI allows a maximum address length of 15 digits. Thus, if 1-digit subaddressing is used, a digit is always available. If, however, 2-digit subaddressing is used, only 13 digits will be available for the address. The network ignores address digits after those actually needed for addressing.

Note: NPSI cannot tell whether the last address digits are part of the address or are added by the calling DTE for CTCP selection purposes.

Defining X.25 Networks

The following 1-digit example shows how you can define subaddressing digits to select a CTCP:

FTPI=SUBD, CTCP=(0,1,2), SUBD=(7,8,9)

The preceding coding produces the following correspondence:

- CTCP 0 is selected if 7 is the last subaddressing digit.
- CTCP 1 is selected if 8 is the last subaddressing digit.
- CTCP 2 is selected if 9 is the last subaddressing digit.

The following 2-digit example shows how you can define subaddressing digits to select a CTCP:

CONNECT=SUBD, CTCP=(0,1,2), SUBD=(07,08,10)

Because the SUBD values have 2-digits, 2 subaddressing digits will be used for CTCP selection. As a result, the following correspondences are established:

- CTCP 0 is selected if 07 is the last subaddressing digit.
- CTCP 1 is selected if 08 is the last subaddressing digit.
- CTCP 2 is selected if 10 is the last subaddressing digit.

Note: Mixing of 1- and 2-digit values on the SUBD parameter is not allowed.

Called Address Extension Facility (AEF) (NPSI V3R4 and Later): The Called Address Extension Facility (facility code X'C9') uses X.213/ISO 8348 addresses to establish routing decisions based on this facility.

The bytes following the facility code (X'C9') give the length of the AEF parameter field in bytes. Table 4-8 shows the meaning of the first two bits.

Table 4-8. Use of Called Address Extension in Bits 0 and 1

Bit 0	Bit 1	Use of Called Address Extension
0	0	To carry a called address assigned according to Recommendation X.213/ISO 8348
0	1	Reserved
1	0	Other (to carry called address <i>not</i> assigned according to Recommendation X.213/ISO 8348 AD2)
1	1	Reserved

For Call-Out

Call-out is performed to a known destination. The LLC type is known before the call is actually established. For GATE and DATE, the Call Request packet for call-out is built entirely by the CTCP.

Use the VTAM PATH statement to specify the LLC type for non-GATE and non-DATE virtual circuits. The VTAM PATH statement is coded in the switched major node entry of the VTAM definition library.

X.25 uses a modified version of the DIALNO or DLCADDR keywords of the PATH statement. The dial number contains various parameters required to perform the call-out function. This keyword format can be found in "PATH Definition Statement" on page 6-8.

X.25 uses the L field of the DIALNO or DLCADDR keywords to determine the LLC type. Table 4-9 on page 4-27 shows the L values, the LLC type created, and the beginning of the CUD field.

Table 4-9. DIALNO or DLCADDR L Values

L	LLC Type	Beginning of the CUD Field
0	Type 0	X'C0'
1	Type 3 (Subarea node)	X'E3' for NPSI X'EB' for X.25 ODLC
2	Type 2	X'C2'
3	Type 3 (Peripheral node)	X'C3' X'C30100008dddwww0000000' for use by an IBM 3710 Controller, where: <i>ddd</i> 3-decimal digit upstream address (USA) of the terminal addressed by this outgoing call <i>www</i> 5-decimal digit ID number of the PU used to map the terminal addressed by the outgoing call X'CB' for X.25 ODLC
5	Type 5	X'01000000'
6	Type 5	X'51000000'
7	Type 5	First positional parameter specified in CUD0 keyword corresponding to an 85 in the CTCP keyword. See "LLC Type and Extended CTCP Selection" on page 4-28 for more information.
8	Type 5	X'81000000'
9	Type 5	X'41000000'
colon (:)	Type 0	First positional parameter specified in CUD0 keyword corresponding to an 80 in the CTCP keyword, followed by ZZZZ. See "LLC Type and Extended CTCP Selection" on page 4-28 for more information.
period (.)	Type 0	First positional parameter specified in CUD0 keyword corresponding to an 80 in the CTCP keyword. See "LLC Type and Extended CTCP Selection" on page 4-28 for more information.

Note: When the call-out is initiated by a CTCP using GATE, NPSI automatically uses LLC type 4.

Call initiation from a DATE CTCP allows you to specify the LLC type within the CALL REQUEST command. The physical circuit is defined to NPSI as being controlled by a DATE CTCP. The DATE CTCP must:

1. Acquire an MCH LU.
2. Select an LLC type.
3. Establish the session to the virtual circuit LU. This is done by any one of the following:
 - NPSI generates a logon request.
 - A user performs LOGON from a remote device.
 - A LOGAPPL keyword is coded on the LU statement in the SMN.

LLC Type and Extended CTCP Selection

If GATE=GENERAL is defined on the MCH, you can create a user-defined table for selecting LLC types by using the CUD0 and CTCP keywords of the X25.MCH statement.

Use the CUD0 and CTCP keyword correspondence to create a one-for-one correspondence between LLC type and CUD0 codes.

To code a one-for-one correspondence, code the CTCP keyword of the X25.MCH statement with the number of subkeywords equal to the number of CUD0 codes in the CUD0 keyword. The *position* of each code determines the LLC type or CTCP number to which it corresponds. You must code the LLC type numbers as the LLC type plus 80, as shown in Table 4-10.

Table 4-10. LLC Numbers

LLC Type	Code	Type
Type 0	80	PCNE -SNA, X.25
Type 2	82	PSH SNA using NIA
Type 3	83	Peripheral node/QLLC SNA/QLLC (Peripheral node)
Type 5	85	PAD non-SNA using PAD

Note: LLC type 4 (GATE) is not selected by code 84 but by a CTCP number (00–27) instead.

Because the CUD0 and CTCP keywords are coded on the X25.MCH statement, you can use them to create a different table for each physical circuit. For call-outs, the value placed in CUD0 can also be user-specified for LLC type 0 and LLC type 5.

The following is an example for coding a user-defined table. You can specify any CUD0 value including a NULL value.

```
CUD0=(00,01,C1,C5,02,FF,03,04,E3,73,61,NULL)
CTCP=(00,01,80,85,85,80,01,02,83,02,85,80)
```

Note: When CUD0 has a NULL value that selects LLC type 0 (PCNE), the default IDNUM is used. The result is in the following user-defined table (Table 4-11).

Table 4-11 (Page 1 of 2). Results of Coding Example for CUD0 Value

CUD0 Parameter Value	CTCP Keyword Value	CTCP or LLC Type Selected	Result
00	00	CTCP 00	
01	01	CTCP 01	
C1	80	LLC 0	C1 goes in the CUD0 field for L= or on call-out.
C5	85	LLC 5	C5 goes in the CUD0 field for L=7 on call-out.

Table 4-11 (Page 2 of 2). Results of Coding Example for CUD0 Value

CUD0 Parameter Value	CTCP Keyword Value	CTCP or LLC Type Selected	Result
02	85	LLC 5	
FF	80	LLC 0	
03	01	CTCP 01	
04	02	CTCP 02	
E3	83	LLC 3	
73	02	CTCP 02	
61	85	LLC 5	
NULL	80	LLC 0	

In addition, default LLC type selection is made *except* for X'00', X'01', X'02', X'03', and X'04', which were overridden within the table.

CTCP Selection Using the Address Extension Facility

GATE CTCP selection support allows NPSI to conform to an OSI environment and its rules. This creates a new mechanism for CTCP selection in GATE by using addresses transmitted in the called address extension facility (AEF) of the Call Setup packets.

The last byte of the AEF facility (X'C9') on the Incoming Call packet is used for CTCP selection in GATE.

When an Incoming Call is received for an MCH specified with AEF, CTCP selection is implemented through the AEF when the following four conditions are met:

- The MCH can be specified without the SUBADDR keyword; or with subaddressing, but no called DTE address in the Call packet; or with subaddressing and the subaddressing digit set for OSI. (It does not select LLC0, LLC2, LLC3, or LLC5.)
- The CUD field in the Call packet is null or begins with X'03010100'.
- The AEF exists in the Call packet.
- A correspondence is found between the last byte of the AEF and a CTCP (for example, when the AEF is specified as a CTCP selector).

V3R8 or Later: CTCP selection by AEF is authorized on MCHs dedicated to X.21 switched connections. The same conditions as for a regular GATE MCH apply except that the CUD field is not checked and that SUBADDR=YES is not allowed.

The existing process to select an LLC/CTCP is used when at least one of these conditions is met:

- The MCH is generated with subaddressing, and the subaddressing digit selects LLC0, LLC2, LLC3, or LLC5.

Defining X.25 Networks

- The CUD field in the Call packet is not null or does not begin with X'03010100'.
- The AEF does not exist in the Call packet.
- Correspondence is not found between the last byte of the AEF and a CTCP. (The AEF byte has not been generated as a CTCP selector.)

System Generation Parameters

To select a CTCP, the following keywords must be added on the X25.MCH statement:

```
AEFBYTE=(aa,bb,...)  aa,bb represent the last byte of AEF field
                      (in hexadecimal) (range 00–FF)
AEFCTCP=(ii,jj,...)  ii,jj are the CTCP numbers (decimal)
                      (range 00–27)
```

Note: The elements in the lists must be coded in 2 digits.

A correspondence is established between the two keywords. For example, CTCP *jj* is selected if the last byte of the AEF is *bb*. Up to 28 CTCPs can be selected in this manner.

The new keywords are compatible with CUD0, CTCP, and SUBADDR keywords. Because this function is supported on GATE, but not fast connect, CONNECT=NO and GATE=GENERAL are mandatory with AEFBYTE and AEFCTCP.

CTCP Definitions

GATE and DATE programming facilities use the CTCP for virtual circuit setup, termination, and control. The following sections outline the network requirements for GATE and DATE and define special requirements for using the GATE fast connect option.

DATE

DATE is used with all virtual circuits that reside on a physical circuit defined as GATE=DEDICAT; consequently, you are unable to:

- Have a physical circuit where *both* DATE and GATE virtual circuits reside
- Bypass the CTCP when performing any CALL REQUEST

NPSI supports DATE and integrated PAD on the same physical circuit.

To generate DATE support, code GATE=DEDICAT on the X25.MCH statement as shown in the following example:

```
X25.MCH  ADDRESS=xxx,
          .
          .
          GATE=DEDICAT
          .
          .
```

When you use DATE, you must code the CALL keyword of the X25.LINE or X25.VC statement in accordance with your PSDN subscription. This allows NPSI to verify that the CTCP is functioning in accordance with the PSDN subscription.

A call-out made by the CTCP is simulated by a call-in to NCP and the access method. Therefore, NPSI forces CALL=IN on all NCP LINE statements created during NPSI generation.

V3R6 and Later: In current DATE MCH processing, NPSI gets the MCH LU name from the BIND, which flows over the LU-LU control session. At X.25 SVC establishment, NPSI builds a pseudo-logon and includes the MCH LU name. This allows an application to relate the virtual resource to its physical owner. To provide CICS users with the same capability, a new command, MLUNAME, carries the assigned MCH LU name and can be used in the LU-LU session.

GATE

Because GATE defines its own LLC type (type 4), GATE cannot reside on the same physical circuit as DATE.

To generate GATE support, code GATE=GENERAL on the X25.MCH statement as shown in the following example:

```
X25.MCH  ADDRESS=xxx,
          .
          .
          GATE=GENERAL
          .
          .
```

To specify the selection of GATE CTCP when the first byte of the CUD field is not X'C4', you must define the following keywords of the X25.MCH statement:

- CUD0
- CTCP
- SUBADDR

When SUBADDR=YES, specify the keywords that select subaddress digits.

V3R6 and Later: In current GATE MCH processing, NPSI gets the MCH LU name from the BIND, which flows over the LU-LU control session. At X.25 SVC establishment, NPSI builds a pseudo-logon and includes the MCH LU name. This allows an application to relate the virtual resource to its physical owner. To provide CICS users with the same capability, a new command, MLUNAME, carries the assigned MCH LU name and can be used in the LU-LU session.

PVC Specifications

PVCs that use GATE can be defined for virtual circuits in the following ways:

- Code GATE=GENERAL on the X25.MCH statement.
- Code LLC=LLC4 on the X25.VC or the X25.LINE statement.
- Code TYPE=PERMANENT on the X25.VC or the X25.LINE statement.

Note: If you use the X25.LINE statement, you must also define the X25.LU and the X25.PU statements.

SVC Specifications

Specifying GATE on an SVC differs for call-in and call-out. For either operation, you must code the GATE=GENERAL keyword on the X25.MCH statement.

Call-In: To perform call-in on a virtual circuit, code the CALL keyword as either CALL=IN (the default value), or CALL=INOUT. You can determine the LLC type through the use of the CUD0 field or the subaddressing field. If you want to control the virtual circuits, select the CTCP by using one of the following:

- Omit CUD0 on the X25.MCH statement.

In this case, there can be only *one* CTCP. This CTCP is chosen if the first byte of the CUD field is:

- NULL
- X'00'
- Any value from X'02' through X'2F'
- X'C4'

- Code CUD0=ALL

This specification allows for two CTCPs. The first CTCP is chosen if the first byte of the CUD field is:

- X'02'
- X'C4'

The second CTCP is chosen if the CUD field has one of the following values:

- NULL
- X'00'
- Any value from X'03' through X'2F'

- Use the correspondence table.

If a CUD0 to CTCP correspondence table is created, the CTCP is selected as specified in the table. Any of the 28 CTCPs can be selected in this manner. Each physical circuit can use a different set of rules for CTCP selection.

In addition to the NPSI generation specifications, the CTCPs must be numbered sequentially according to the logical unit associated with the physical circuit. The CTCP numbers range from 00—27.

When you use a table, you can define specific CTCPs to establish sessions with the appropriate physical circuit logical unit. All LU-LU sessions between the CTCPs and logical units do not have to be continuously active. Additional LU-LU sessions can be established as network traffic requires.

When subaddressing is used to select the LLC, the first CTCP is selected if the subaddressing digit of the Call packets matches one of the values specified for the LLC4 keyword of the X25.MCH statement. If it does not match any of these values, NPSI uses the CUD0 of the Incoming Call packet to select the CTCP.

Call-Out: When GATE is used and a call-out is requested by the CTCP, NPSI simulates a call-in to the NCP and VTAM. Therefore, all the virtual circuits that can be used for GATE call-out must be declared as CALL=IN or CALL=INOUT at generation. Because all virtual circuits on an MCH are not dedicated to operating through GATE, NDF passes the coded CALL keyword (without change) to the NCP and VTAM.

When you want to reserve a certain number of virtual circuits for non-GATE call-outs, code CALL=OUT for this number of virtual circuits.

When the CTCP requests call-out on a physical circuit coded as using GATE, the call must use a type 4 virtual circuit and must meet the following two specifications in order to be successful:

- Include LLC4 in the LLCLIST keyword of the X25.MCH statement.
- Code CALL=IN or CALL=INOUT on the X25.LINE or X25.VC statement to ensure the availability of a virtual circuit.

Fast Connect Option

MCH Keywords: Fast connect support requires that you add the CONNECT keyword on the X25.MCH statement at the time of generation. To select the CTCP, use one of the four CONNECT options:

CONNECT=NO	CONNECT=NO is the default and results in a non-fast connect MCH.
CONNECT=YES	When CONNECT=YES is specified, <i>neither</i> the CUD0 nor the SUBD keywords can be used. Only a single CTCP can be used with CONNECT=YES.
CONNECT=CUD0	When CONNECT=CUD0 is specified, the first byte of the CUD field (CUD0) is used to select the CTCP. This keyword is paired with the CTCP keyword to create a correspondence table between the positional parameters specified within the CUD0 and CTCP keywords. This specification allows you to define up to 28 different CTCP destinations.
CONNECT=SUBD	When CONNECT=SUBD is specified, the subaddressing digits are used as the selection method for CTCP destination. In the case of fast connect, the <i>last</i> subaddressing digits determine the correct CTCP destination. Like CUD0, this keyword is paired with the CTCP keyword to create a correspondence table. For examples, refer to “Using Subaddressing for CTCP Selection (NPSI V3R4 and Later)” on page 4-25.

Notes:

1. When subaddressing is used for CTCP selection with fast connect, but a subaddressing digit is not found in the correspondence table, the CUD0 field *cannot* be used as an alternative connection method, as it is in GATE. In this case, the call will be cleared.
2. When subaddressing is used for CTCP selection, if an Incoming Call packet is received without a called DTE address (no subaddressing digit), the call is treated as if there was a subaddressing digit of 0.

For each different CTCP specified in the CTCP keyword, a set of additional keywords can be specified in relation to the CTCP number. Thus, CTCP-0 is specified as the first positional parameter on each of these keywords, CTCP-1 is the next positional parameter, and so on through CTCP-27. A maximum of 28 parameters can be specified on the following keywords:

LUNAME	Specifies the MCH LU names corresponding to each CTCP. These secondary logical units (SLUs) are in session with the appropriate CTCP.
--------	---

Defining X.25 Networks

LOGAPPL	Specifies the application name to which the LU is attached once the LU is successfully activated. This keyword places the VTAM-only keyword on the LU statements created by the X25.MCH statement and on the LUs created for the SNA resources used to map the SVCs related to the CTCP.
MODETAB	Defines the logmode entry table used between the MCH LUs defined in the LUNAME keyword and the CTCPs defined in the LOGAPPL keyword.
TAXUNIT	Specifies the number of characters that define a network tax unit that can be used for billing purposes. Valid values for this keyword are 0, 32, 64, and 128. When this keyword is specified and either a CLEAR command, CLEAR CONFIRMATION command, or information/error report is sent to the CTCP, NPSI appends the billing information to the specified command. The use of billing units as statistical data affects panels NPDA-51F and NPDA-53F in NetView. In these panels, both the TRANSMISSION TOTAL and RECEIVE TOTAL can reflect billing units and not packet totals.
VMODTAB	Defines the logmode table entry used for the sessions between the virtual circuits and the CTCPs. These virtual circuits are defined in the X25.FCG statement and the X25.VC statement.

FCG Keywords: Fast connect also requires that specific keywords be used in the X25.FCG statement. If more than one destination CTCP is attached to a particular MCH, an X25.FCG statement must be coded for each CTCP. This statement also defines the number of SNA sessions that will be allocated for fast connect use. Use the following keywords to achieve this:

CTCPNO	Specifies which CTCP is being defined by the statement. Valid values range from 0–27.
QTY	Defines the number of SNA sessions created by this statement. The maximum value of this keyword is 4096.

The X25.FCG statement also defines the fast connect LINE, PU, and LU names with the following five keywords:

PRFLINE	Defines the prefix used for all LINES.
PRFPU	Defines the prefix used for all PUs.
PRFLU	Defines the prefix used for all LUs.
SUFFIX	Defines the lower limit of the resource numbers.
HEXNAME	Defines whether the suffix incrementation is in hexadecimal or decimal; the default is YES.

The resource name is created by appending the suffix to the appropriate prefix. The length of the names is constant and is defined by the sum of the length of PRFLINE, PRFPU, or PRFLU, added to the length of the SUFFIX parameter.

Note: Because there is an 8-character limitation on labels, as the prefix increases, the suffix size decreases, reducing the number of resources that can be defined. PRFLINE, PRFPU, PRFLU, and SUFFIX have no default. Either choose the NPSI

default naming convention or specify the names using the PRFLINE, PRFPU, PRFLU, and SUFFIX keywords. The specifications cannot be mixed. X25.FCG also creates a GROUP statement for each instance of X25.FCG.

The following X25.FCG keywords are used to define specifications that sift down to the LINE, PU, and LU statement levels:

- MAXDATA
- ISTATUS
- NCPGRP
- OWNER
- SPAN
- ANS

Figure 4-3 on page 4-36 shows an example of a fast connect configuration. To create this environment, you can use the following NPSI keywords as an example:

```
X25.MCH      ADDRESS=XXX,
             CONNECT=CUD0,
             LOGAPPL=(CTCP1,CTCP2),
             CUD0=(00,09,01),
             CTCP=(00,01,00)

X25.LCG      LCGN=4

X25.VC       LCN=(1,100),
             VCCINDX=01,
             TYPE=S,
             CALL=IN

X25.FCG      QTY=80,
             CTCPNO=0,
             MAXDATA=2580,
             PRFLINE=XL12,
             PRFPU=XP12,
             PRFLU=XL12,
             SUFFIX=01

X25.FCG      QTY=70,
             CTCPNO=01,
             MAXDATA=2580,
             PRFLINE=XL13,
             PRFPU=XP13,
             PRFLU=XL13,
             SUFFIX=01
```

Note: SNA resources are defined using the X.25 FCG statement. Virtual circuits are defined using the X25.VC statement.

Defining X.25 Networks

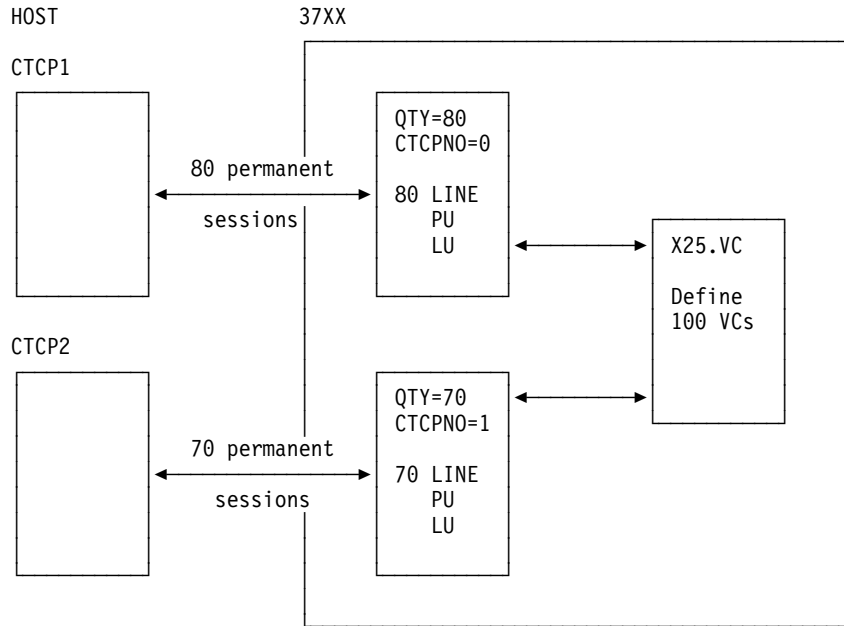


Figure 4-3. Fast Connect with Two CTCPs for One Physical Circuit

QOS (NPSI V3R4 and Later)

NPSI support for the QOS priority of data allows you to prioritize the data traffic to the network on one or more virtual circuits on the MCH.

The DEFPRI keyword defines the default for QOS priority, and is defined on the X25.MCH, X25.LINE, and X25.VC statements.

X25.MCH Statement

When specified on the X25.MCH statement, DEFPRI defines the default priority value assigned to all virtual circuits attached to this MCH. The QOS priority value is assigned to:

- Non-GATE peripheral node traffic (PVCs and SVCs)
- GATE leased traffic, if the DEFPRI parameter is not overridden on the X25.VC or X25.LINE statements
- GATE switched traffic, whose QOS priority facility is not present in Call Setup packets
- GATE switched traffic, whose QOS priority facility value is reserved (X'0F' to X'FE'), or unspecified (X'FF')

For the X25.MCH statement, the parameters for the DEFPRI keyword are:

```
X25.MCH DEFPRI=NO
                LOW
                NORMAL
                HIGH1
                HIGH2
```

The default value for the DEFPRI keyword on the X25.MCH statement is NO, which means priority processing is not done on this MCH.

Notes:

1. You can code DEFPRI on the X25.MCH statement only if GATE=GENERAL is also coded on the X25.MCH.
2. You can code OSIPRI on the RFAC keyword of the X25.NET statement to eliminate processing for all SVCs.

See Chapter 5, “Resource Definition Reference” for more information about coding the DEFPRI keyword on the X25.MCH statement.

X25.LINE Statement

When specified on the X25.LINE statement, DEFPRI defines the QOS priority value for GATE PVCs.

The parameters for the DEFPRI keyword on the X25.LINE statement are:

```
X25.LINE DEFPRI=NO
          LOW
          NORMAL
          HIGH1
          HIGH2
```

Notes:

1. DEFPRI is authorized on the X25.LINE statement only when TYPE=P and LLC=4 are coded.
2. If you code DEFPRI=LOWINORMALIHIGH1IHIGH2 on the X25.MCH statement, you cannot code DEFPRI=NO on the X25.LINE statement. If you code DEFPRI=NO on the X25.MCH statement (or if DEFPRI is not coded on the MCH) you cannot code DEFPRI=LOWINORMALIHIGH1IHIGH2 on the X25.LINE statement.

The default value for the DEFPRI keyword of the X25.LINE statement is the value coded on the X25.MCH statement.

See Chapter 5, “Resource Definition Reference” for more information about coding the DEFPRI keyword on the X25.LINE statement.

X25.VC Statement

When specified on the X25.VC statement, DEFPRI defines the QOS priority value for GATE PVCs.

The parameters for the DEFPRI keyword on the X25.VC statement are:

```
X25.VC DEFPRI=NO
          LOW
          NORMAL
          HIGH1
          HIGH2
```

Defining X.25 Networks

Notes:

1. DEFPRI is authorized on the X25.VC statement only when TYPE=P and LLC=4 are coded.
2. If you code DEFPRI=LOWINORMALIHIGH1IHIGH2 on the X25.MCH statement, you cannot code DEFPRI=NO on the X25.VC statement. If you code DEFPRI=NO on the X25.MCH statement (or if DEFPRI is not coded on the MCH) you cannot code DEFPRI=LOWINORMALIHIGH1IHIGH2 on the X25.VC statement.

The default value for the DEFPRI keyword of the X25.VC statement is the value coded on the X25.MCH statement.

See Chapter 5, "Resource Definition Reference" for more information about coding the DEFPRI keyword on the X25.VC statement.

PVC

For PVCs, the QOS priority of data is defined in the following manner:

- For GATE PVCs, the priority facility is defined at system generation time on the X25.LINE or X25.VC statement. If neither of these is defined, the value coded on the X25.MCH is used.
- For peripheral node PVCs, the priority defined at system generation time on the X25.MCH is used.
- For subarea node PVCs, the highest priority (HIGH2) is used.

SVC

For NPSI GATE SVCs, the QOS priority of data is defined at call setup in the following manner:

- Call-out

The Call Request packet (with or without priority facility) is passed unchanged from the CTCP to the network through NPSI. The selected value, if any, is saved by NPSI from the Call Connected packet received from the network.

- Call-in

The Incoming Call packet (with or without the priority facility) is passed unchanged from the network to the CTCP through NPSI. The selected value, if any, is saved by NPSI from the CALL ACCEPTED command passed by the CTCP.

If the facility field is not present in Call packets, a default priority as specified on the X25.MCH statement is used.

For other peripheral node SVCs, NPSI ignores the priority facility, even if it is present in the Call packets. The default priority specified on the X25.MCH is always used.

For subarea node SVCs, NPSI ignores the priority facility, even if it is present in the Call packets. The highest priority (HIGH2) is used.

Fast Transaction Processing Interface (NPSI V3R4 and Later)

The X25.CPL statement is used to define FTPI CTCP characteristics. X25.CPL has the following format:

```
X25.CPL ALTCTCP=...,
        CTCPNO=...,
        MAXTIME=...,
        MINDATA=...,
```

where:

ALTCTCP	Specifies the alternative CTCP identification, and can specify up to 27 alternative CTCPs if the primary CTCP is in a non-responsive state. This value must be specified on another X25.CPL statement.
CTCPNO	Specifies the CTCP identification, and can specify up to 27 individual CTCPs assigned by the X25.CPL statement. Each X25.CPL statement must have a unique CTCPNO keyword.
MAXTIME	Specifies the maximum time, in tenths of seconds, to wait before sending a PIU to the CTCP.
MINDATA	Specifies the blocking factor (the <i>maximum</i> number of packets that can be multiplexed into one PIU sent to the FTPI CTCP).

Note: Code one X25.CPL statement for each FTPI CTCP and GATE=GENERAL on the X25.MCH statement.

Code the FTPI keyword on the X25.MCH statement to initiate FTPI processing. The format for the FTPI keyword is:

```
FTPI = {NO|YES|CUD0|SUBD|PVC}
```

where:

NO	Specifies the default and that FTPI is not used on this MCH.
YES	Specifies that FTPI is used on this MCH. Code FTPI=YES when there is only one CTCP.
CUD0	Specifies that FTPI with multiple CTCPs is used on this MCH, and the CTCP is selected through CUD0. The CUD0 and CTCP keywords must be coded together (with the same number of subvalues) to create a correspondence table between two positional operands.
SUBD	Specifies that FTPI with multiple CTCPs is used on this MCH, and the CTCP is selected through the subaddressing digits. The SUBD and CTCP keywords must be coded together (with the same number of subvalues) to create a correspondence table between two positional operands. For examples, refer to "Using Subaddressing for CTCP Selection (NPSI V3R4 and Later)" on page 4-25.
PVC	V3R8 or Later: Specifies that only PVCs are defined for this FTPI MCH. The CTCP keyword must be coded with the identification of the authorized CTCP(s).

Defining X.25 Networks

The FTPI CTCP selection mechanism is identical to FC and uses the same keywords.

You can code FTPI on the X25.MCH statement when the following conditions are met:

- At least one X25.CPL statement is coded.
- GATE=GENERAL is coded on the X25.MCH statement.
- LLC type 4 is the only authorized LLC for the virtual circuits of the MCH.

When you code FTPI on the X25.MCH statement, all of the virtual circuits on this MCH are dedicated to FTPI.

Note: VTAM keywords (for example, LOGAPPL and MODETAB) are not authorized for FTPI MCHs. The VTAM keywords coded on the X25.CPL statement are used for the corresponding CTCP number.

NCP V7R2 Only: NPSI APAR IR27307 must be applied for FTPI to work with NCP V7R2.

PAD Support

PAD support is defined on the X25.MCH statement using the PAD keyword. NPSI provides two types of PAD support:

- Integrated PAD
- Transparent PAD

Integrated PAD support is used to define a PAD that conforms to CCITT Recommendation X.28. To define integrated PAD support, you must code the PAD keyword as PAD=INTEG on the X25.MCH statement.

Integrated PAD support also allows applications to take advantage of NPSI password protection. To do this, the application must use the inhibit presentation (INP) or the enable (ENP) presentation characters. Hexadecimal codes for these are:

```
INP=24  
ENP=14
```

Note: INP and ENP can be displayed on some terminals as a false message. You can avoid this condition by coding PWPROT=YESWOCC on the X25.MCH statement, which causes NPSI to suppress the transmission of these characters.

V3R5 and Later: For the integrated PAD function, NPSI V3R5 and later releases make sending of the READ PAD PARAMETER command optional at call setup time. In some instances, when the READ PAD PARAMETER command is sent, some data may be lost from the DTE. You can avoid this condition by coding RPADPARM=NO on the X25.MCH statement. For more information about coding the RPADPARM keyword, see Chapter 5, "Resource Definition Reference."

Use transparent PAD support for all other types of PADs, except for the SDLC PAD. You should use transparent PAD support if you require more PAD control than is provided with the integrated PAD. To define transparent PAD, you must code the PAD keyword as PAD=TRANSP on the X25.MCH statement.

The default on the PAD keyword is PAD=NO. If you code NO or leave the PAD keyword off, PAD support is not available for virtual circuits on the physical circuit.

Note: LLC5 must be included in the LLCLIST to use PAD SVCs.

Generation of PAD Parameters

You can generate one or more strings of PAD parameters by using the X25.PAD statement. You can select one parameter for each X25.PAD statement by coding an index value on the PADINDX keyword on the X25.MCH statement.

PCNE-to-PCNE Considerations

If you specify a PU associated with a type 0 virtual circuit, The remote identification number generated in the Call Request packet is in CUD field bytes 1—3. This remote identification number, received from an Incoming Call packet by the remote DTE, will equal the ZZZZZ field plus 1.

The only relationship between two SNA domains is made by the remote identification number specified in the Call Request and Incoming Call packets.

The IDNUM keywords coded in the two domains need to be correlated so that it is clear which PU of one domain calls which PU of the other domain. Code your PU and PATH statements as illustrated in the following example:

SNA Domain 1			SNA Domain 2		
PU11	PU	IDBLK=003, IDNUM=90001,	PU22	PU	IDBLK=003, IDNUM=90002,
		.			.
		.			.
		.			.
PA11	PATH	DIALNO...*90001,	* No PATH (PU is used for call-in)		
		.			
		.			
		.			

The ZZZZZ field of PU11 is used as IDNUM for PU11 and is incremented by 1 to be used as IDNUM for calling in PU22.

Multichannel Link Compatibility

Table 4-12 shows the compatibility of GATE, DATE, integrated PAD, transparent PAD, and PCNE on the same multichannel link for NPSI.

NPSI allows both integrated PAD and DATE to coexist on the same multichannel link.

Table 4-12. Enhanced Multichannel Link Compatibility

	GATE	DATE	Integrated PAD Support	Transparent PAD Support	PCNE
GATE	N/A	No	Yes	Yes	Yes
DATE	No	N/A	Yes	Yes	Yes
Integrated PAD Support	Yes	Yes	N/A	No	Yes
Transparent PAD Support	Yes	Yes	No	N/A	Yes
PCNE	Yes	Yes	Yes	Yes	N/A

Chapter 5. Resource Definition Reference

How Definition Statement and Keyword Information Is Presented	5-2
Resource Naming Conventions	5-2
Coding Conventions Used in This Chapter	5-6
BUILD Definition Statement	5-8
GENEND Definition Statement	5-12
SWMMT Definition Statement	5-13
X25.CPL Statement (NPSI V3R4 and Later)	5-14
X25.END Statement	5-21
X25.FCG Statement	5-22
X25.LCG Statement	5-30
X25.LINE Statement	5-31
X25.LU Statement	5-41
X25.MCH Statement	5-45
X25.NET Statement	5-80
X25.OUFT Statement	5-87
X25.PAD Statement	5-89
X25.PU Statement	5-90
X25.TRAN Statement	5-95
X25.VC Statement	5-97
X25.VCCPT Statement	5-111

This chapter presents the resource naming conventions and resource definition statement formats in alphabetical order, uses tables to illustrate the statement format, and describes the keywords for each statement. For clarity and retrievability, each statement section starts on a new page.

Read “Planning for NPSI and X.25 ODLC Migration” on page 3-49 to learn about changes in the X.25 generation process.

Read “Relationship Between NCP and NPSI” on page 2-2 to learn about the new X.25 ODLC support provided by the Network Control Program (NCP) starting with V7R4.

All NPSI V3R8 and later keywords that apply to X.25 LLC3 resources also apply for X.25 ODLC unless stated otherwise.

How Definition Statement and Keyword Information Is Presented

This chapter contains detailed information on how to code the definition statements and keywords that define X.25 resources. There is a section for each X.25 definition statement; these sections appear in alphabetical order. Each section contains the following subsections:

Usage:

Lists the functions or resources for which the definition statement is used.

Position:

Indicates where the definition statement must appear, relative to other definition statements in your generation definition.

Format and Keywords:

Shows the format for coding the statement and provides a table that lists the keywords described under the definition statement and the VTAM keywords you can code on the definition statement. The VTAM keywords must appear in the generation definition that is the input to the VTAM initialization process, but they are not required in the definition you use to generate X.25 resources. The X.25 generation procedure does not check the VTAM keywords for proper syntax or verify that any related keywords are present or absent. For more information about VTAM keywords, see the *VTAM Resource Definition Reference*. The second part of this subsection describes each X.25 keyword in detail; the keyword descriptions appear in alphabetical order.

Resource Naming Conventions

X.25 resource names are used by VTAM, the NetView program, and teleprocessing subsystems, such as CICS or IMS, during the operation of the network system.

X.25 resource names can be created in three ways:

- NDF creates default resource names during the generation process if you do not specify a label on the resource statement. Be sure that your own user-specified names are not duplicated by the default resource names built by NDF.
- You can override the default and code your own user-specified names on the X25.LINE, X25.MCH, X25.PU, and X25.LU definition statements.
- You can specify resource names generically on the X25.FCG and X25.VC definition statements.

All resource names built by default during X.25 generation use the letter specified by the value of the X25.PREFIX keyword on the NCP BUILD statement. This is by default, an X. See “BUILD Definition Statement” on page 5-8 for information about replacing X with a user-specified letter.

NDF builds the resource names for the following:

- Physical circuit:

```
pLxxx LINE
pPxxx PU
pUxxx LU
or
pUxxxLnn (for any LU other than the first, if more than one LU
           is defined for the MCH GATE)
```

where:

p Is the X25.PREFIX keyword of the BUILD statement (default=X).

xxx Is the physical circuit line address as coded in the X25.MCH ADDRESS keyword (1–3 digits).

nn Is the number of the LU (for example 01 for LU1, 02 for LU2 and so on, up to 27). The first LU (LU number 00) is named pUxxx.1.

- GROUP statement created by X25.MCH

Each X25.MCH statement creates a unique GROUP statement:

```
pNETnm
```

where:

p Is the X25.PREFIX keyword of the BUILD statement (default=X).

n Is the position of the previous X25.NET statement within the X.25 generation source statements.

m Is the rank of this MCH within the previous X25.NET statement.

- Virtual circuit:

```
pLxxx yzz LINE
pPxxx yzz PU
pUxxx yzz LU
```

where:

p Is the X25.PREFIX keyword of the BUILD statement (default=X).

xxx Is the physical circuit line address as coded in the X25.MCH ADDRESS keyword (1–3 digits).

y Is the logical channel group number (in hexadecimal). of the current virtual circuit.

zz Is the SNA resource identifier (in hexadecimal)

Note: The same defaults are generated for LINE, PU, and LU. There is no distinction between the X25.VC, X25.LINE, X25.PU, or X25.LU statements relative to the generation of these defaults.

Resource Definition Reference

- GROUP statement created by X25.LINE or X25.VC statement:

p25Pxxxx for PVCs
p25Sxxxx for SVCs

where:

- p Is the X25.PREFIX keyword of the BUILD statement (default=X).
- xxx Is the physical circuit line address as coded in the X25.MCH ADDRESS keyword (1–3 digits).
- y Is the letter specifying the group sequence on the physical circuit. The first group will be assigned an A, the second group a B, and so on.

- SVCSC:

pLxxxxvvv LINE
pPxxxxvvv PU

where:

- p Is the X25.PREFIX keyword of the BUILD statement (default=X).
- xxx Is the physical circuit line address as coded in the X25.MCH ADDRESS keyword (1–3 digits).
- vvv Is the number of the SVCSC line or PU within the MCH expressed in base 20, with letters G–Z used to represent the numbers 0–19, respectively. For example, the first name is pLxxxGGH; the second name is pLxxxGGI, and so on.

Note: In the naming conventions shown previously, xxx is the physical line address for non-HPTSS lines coded on the ADDRESS keyword of the X25.MCH statement. For HPTSS lines, xxx represents A24 to A43, and valid ADDRESS keyword values range from 1024 to 1043. For CSS lines, xxx represents the last three digits of the MCH address expressed in base 20 with the letters G–Z used to represent the numbers 0–19, G standing for 0, H for 1... Z for 19. For example, the first valid CSS address representation 2112 will be represented by the characters GLS which corresponds to 112 converted in base 20 then translated with the G - Z table (G is the first character of the table, L the sixth, and S the thirteenth).

Another example: for the highest valid CSS address representation 2750, the characters will be HXQ.

- X25.FCG statement:

pLFmnggg LINE
pCFmnggg SERVICE
pPFmnggg PU
pUFmnggg LU

where:

- p Is the X25.PREFIX keyword of the BUILD statement (default=X).
- m Is the number of the X25.MCH with CONNECT=CUD0 or CONNECT=SUBD. This single alphanumeric number ranges from 0–9 and A–Z. The number of MCHs with fast connect is limited to 36.
- n Is the number of the CTCP defined within the MCH. This number is then converted into a single alphanumeric in the range 0–9 and A–R.
- ggg Is the SNA resource number in hexadecimal.

- GROUP statement created by X25.FCG statement:

Each X25.FCG statement creates a unique GROUP statement.

```
p25PGmnj  GROUP
```

where:

- p Is the X25.PREFIX keyword of the BUILD statement (default=X).
- m Is the number of the X25.MCH with CONNECT=CUD0 or CONNECT=SUBD. This single alphanumeric number ranges from 0–9 and A–Z. The number of MCHs with fast connect is limited to 36.
- n Is the number of the CTCP defined within the MCH. This single alphanumeric number ranges from 0–9 and A–R.
- j Is an alpha character in the sequence A–Z.

- X25.CPL statement

X25.CPL is an optional statement, and must be coded before the first X25.NET statement of the generation deck.

```
p25CPnnj  GROUP
p25CPLnn  LINE
p25CPCnn  SERVICE
p25CPPnn  PU      (or user coded)
p25CPUnn  LU      (or user coded)
```

where:

- p Is the X25.PREFIX keyword of the BUILD statement (default=X).
- nn Is the CTCP number corresponding to this CPL resource.
- j Is an alpha character in the sequence A–Z.

In addition to NDF-generated names, you can specify unique rules for name creation using the following keywords:

- PRFLINE
- PRFPU
- PRFLU
- SUFFIX
- HEXNAME

The following examples show coding for the keywords PRFLINE, PRFPU, PRFLU, and SUFFIX, and the names that are generated. The first example is:

```
X25.VC  LCN=(1,100),TYPE=S,OUFINDX=1,VCCINDX=1,
        PRFLINE=AL1,PRFPU=AP1,SUFFIX=001,HEXNAME=NO
```

The names generated are:

```
AL1001 to AL1100 for the LINES
AP1001 to AP1100 for the PUs
(Suffix from 001 to 100 in decimal.)
```

The second example is:

```
X25.VC  LCN=(1,100),TYPE=P,VCCINDX=1
        PRFLINE=AL1,PRFPU=AP1,PRFLU=AU1,SUFFIX=001
```

The names generated are:

```
AL1001 to AL1064 for the LINES
AP1001 to AP1064 for the PUs
```

Resource Definition Reference

AU1001 to AU1064 for the LUs
(Suffix from 001 to 064 in hexadecimal.)

Note: To simplify X.25 resource identification in the host processor, use the NDF default naming convention. In NPSI and X.25 ODLC, entering duplicate names and values is not allowed for either resource names or concatenated PRFxx names and SUFFIX values.

NPSI uses macros to describe its control blocks during NCP stage 2 generation. Because these macros are prefixed with BAK, you cannot use BAK to define other macros.

Also, because the entry points within the NPSI load module begin with the prefix BAL, you cannot use BAL to define other entry points in any customized NPSI code.

Note: It is recommended that you choose resource names that do not begin with X even if the value coded on the X25.PREFIX keyword of the NCP BUILD statement is not X. NPSI and X.25 ODLC generate various control blocks that begin with X, and use of this prefix in naming other resources could result in duplicate names. You should likewise not use the X25.PREFIX value as the beginning of resource names due to the possibility of duplicates.

Coding Conventions Used in This Chapter

The following conventions are used throughout this chapter:

- Capital letters represent values you code directly, without changing.
- Small letters represent variables for which you must supply values.
- Commas separate keywords and the keyword values.
- Brackets [] enclose *optional* or *conditional* keywords and symbols.

An *optional* keyword can be coded or omitted, independent of other keywords. If certain keywords are omitted, default values will be used. A keyword's default value is always given as one of the choices.

A *conditional* keyword can be coded or omitted, depending on how you code (or omit) other keywords in the same or other definition statements.

Within the definition statement description, required keywords appear first in alphabetical order. Then the conditional or optional keywords follow in alphabetical order.

- Braces { } indicate that you must choose from the enclosed values.
- OR symbol | indicates that you must choose from the values on either side of the symbol.
- An underlined value represents the default value of the keyword. X.25 generation uses that value if you omit the keyword.

Note: Some keywords show multiple defaults. This indicates that the defaults are different for different releases or for different combinations of other keywords. The text describing these keywords clarifies the default in particular situations.

- Parentheses () must enclose a sequence of values coded for one keyword.

- Single quotation marks must enclose a character string if that character string can be confused with a value for a keyword. Enclosing the character string in quotation marks lets you use certain names as symbols that you would be prevented from using otherwise.
- Numbers have a period between the whole and fractional portions of the numeral. For example, 1.5 denotes one and a half.

In the syntax lists at the beginning of each definition statement, each required keyword, except the last, is followed by a comma. Similarly, each optional keyword, after the last required keyword, is preceded by a comma. However, if the lists consist of optional keywords only, each optional keyword, except the first keyword, is preceded by a comma. These commas are shown to emphasize that you must code a comma between each keyword, but not after the last keyword or before the first keyword of each definition statement.

The X.25 generation procedure does not check NCP, NetView, or VTAM keywords for proper syntax. The generation process also does not verify that any related keywords are absent or present. For the meaning of the keywords, see the appropriate NCP, NetView program, or VTAM installation books.

BUILD Definition Statement

The X25BUILD statement is no longer a valid NPSI definition statement. Information provided by this macro must now be specified on the NCP BUILD statement. If you code an X25BUILD statement, it is flagged with a severity 4 warning message. No keyword processing occurs for the statement found in error.

Several keywords previously coded on the X25BUILD statement are now coded by using equivalent keywords on the NCP BUILD statement. However, some keywords do not have a corresponding NCP keyword. See Appendix C, “Statement and Keyword Changes,” for a complete list of X25BUILD keywords.

Former keywords must be modified before they are added to the NCP BUILD statement. To do this, add the prefix X25. to each keyword to designate it as a unique X.25 keyword. For example, IDNUMH is now coded as X25.IDNUMH.

Note: NCP Version 5 and later releases require that at least one NCP LINE, PU, and LU statement be defined in an X.25 generation before any user LINE statement is defined. A dummy LINE can be defined if ISTATUS=INACTIVE is specified. See Appendix B, “System Generation Examples,” for an example of coding NCP GROUP, LINE, PU, and LU statements.

NPA

If you want to use NPM performance or accounting, you must specify YES on the NPA keyword. See the *VTAM Resource Definition Reference* for more information about the NPA keyword.

CSSTIER

If the IBM IBM 3746 Model 900 is used, this keyword is required. See *NCP, SSP, and EP Resource Definition Reference* for more information.

SESSACC

To perform NPM accounting, you must code YES,ALL on the SESSACC keyword. If you also want to have session accounting for non-X.25 resources, you can code YES,*thresholds* on the SESSACC keyword.

See the *VTAM Resource Definition Reference* for more information about the SESSACC keyword.

X25.BYTTHROD (X.25 ODLC and NPSI V3R4 and Later)

Specifies the maximum byte threshold for NPM accounting. The X25.BYTTHROD values range from 65 536 to 2 147 483 646 with no default value.

Note: Do not code X25.BYTTHROD if the ACCOUNT keyword is not requested on any of the X25.MCH statements.

X25.IDNUMH

Specifies the first digit of the IDNUM for PUs used to map non-SNA resources. Code the IDNUMH value in hexadecimal. The default value is 0; valid values range from 0–F.

X25.MAXPIU

Specifies the maximum length for inbound PIUs. Code the length in kilobytes or in bytes. The X25.MAXPIU value ranges from 2K–1000K for kilobytes and 1296–1 024 000 for bytes.

This keyword is ignored for X.25 ODLC.

Note: X25.MAXPIU should not exceed (255 × the NCP buffer size). This is the default value.

X25.MCHCNT

Specifies the number of physical circuits defined in the X.25 generation statements. Code X25.MCHCNT in decimal. The default value is 1; valid values range from 1–4095.

X25.MMODULO (X.25 ODLC and NPSI V3R7 and Later)

Specifies the modulo value at frame level used by link access procedure balanced (LAPB). Valid values are 8 or 128; 8 is the default value.

X.25 ODLC and NPSI V3R8 and Later:

For X.25 lines connected through the CSS, the following rules apply:

A CLP can support up to four X.25 modulo 128 lines per pair of LIC slots, compared to 60 lines of another type. Each X.25 modulo 128 line reduces the total connectivity by 15 for the other types of lines supported sharing the same CLP. For example, if one X.25 modulo 128 line is defined on a CLP (with two LICs), up to 45 non-X.25 modulo 128 lines such as SDLC, frame relay, or X.25 modulo 8 can also be defined.

X25.MWINDOW

Specifies the frame window size used by LAPB.

V3R6 and Earlier:

Valid values range from 1–7.

X.25 ODLC and NPSI V3R7 and Later:

Valid values depend on the values specified for the MMODULO keyword:

- If you do not specify a value for MMODULO, or if you specify MMODULO=8, the valid values are from 1–7.
- If you specify MMODULO=128, the valid values are from 1–127.

Note: MWINDOW applies to all X25.MCH statements that follow unless the X25.MCH statement overrides the X25.MWINDOW.

X25.OUTIDNUM

Specifies whether default IDNUMs are generated for non-SNA SVCs defined for callout only on MCHs for which the DATE function is not used. Code NO to skip default IDNUM generation. The default is YES.

X25.PAHINDX

Specifies the largest INDEX value coded in the X25.PAD statements. Valid values range from 1–99; the default value is 1.

X25.PREFIX

Specifies the first letter for default resource names. Code the X25.PREFIX value as an alphabetic character that ranges from A–Z, or #, or @. The default value is X.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names.

X25.SEGTHROD (X.25 ODLIC and NPSI V3R4 and Later)

Specifies the maximum segment threshold for NPM accounting. The X25.SEGTHROD valid values range from 65 536 to 2 147 483 646 with no default value.

Note: Do not code X25.SEGTHROD if the ACCOUNT keyword is not specified on any of the X25.MCH statements. Each segment is defined as 64 bytes.

X25.SNAP (NPSI V3R4 and Earlier)

Sets the NPSI SNAP trace facility during generation.

Code X25.SNAP=YES|1000|2000|3000|4000 if you want to include the Snap trace facility. YES or 1000 causes a 1000 entry SNAP area to be included, coding 2000 generates a 2000 entry SNAP area, and so on. When coding these values, the NPSI SNAP trace facility starts automatically.

Code X25.SNAP=NO if you do not want the X25.SNAP trace facility included, that is, you do not intend to use the SNAP trace facility. (The default value is NO.) See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for information about how to use the SNAP trace facility with NPSI.

X25.SNAP (NPSI V3R5 and Later)

X25.SNAP sets the NPSI SNAP trace facility during generation.

Code X25.SNAP=YES|1000|2000|3000|4000,INACT when you want the Snap trace facility included but not active. The INACT value is not allowed when NO is coded. YES or 1000 causes a 1000 entry SNAP area to be included, coding 2000 generates a 2000 entry SNAP area, and so on. When coding these values, the NPSI SNAP trace facility is automatically started. When X25.SNAP=YES|1000|2000|3000|4000 and INACT is not coded, SNAP trace starts automatically.

Code X25.SNAP=NO if you do not want the X25.SNAP trace facility included, that is, you do not intend to use the SNAP trace facility. (The default value is YES,INACT.) See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for information about how to use the SNAP trace facility with NPSI.

X25.USGTIER

Specifies the usage tier where this NPSI load module will operate.

The value must be less than or equal to the usage tier licensed with this NPSI. To determine the ordered usage tier, look at your IBM licensing agreement. The default value is 1; valid X25.USGTIER values are; 1, 2, 2.5, 3, 4, and 5.

If you specify a value that exceeds the installed feature level, the NPSI generation will fail. The X25.USGTIER value specified for NPSI generation must *not* be less than the usage tier value specified by the USGTIER keyword of the NCP BUILD statement.

If you specify an X25.USGTIER value that is less than the USGTIER value specified for the associated NCP, the NPSI generation will fail. The MCHs that can be generated for NPSI depend on the NCP USGTIER. The MCH addresses cannot exceed the USGTIER coded for the NCP.

An example of how to code the usage tier parameters is as follows:

```
BUILD USGTIER=3,      NCP usage tier
      X25.USGTIER=3,  NPSI usage tier
      .               Other parameters
      .
      .
```

This keyword is ignored for X.25 ODLIC.

GENEND Definition Statement

If the NPM accounting function is specified on at least one X.25 MCH statement (ACCOUNT keyword), the X.25 Accounting routine name(s) may either be coded on the UACCTNG keyword of the GENEND statement or is defaulted by the NDF process. The following describes how to code UACCTNG on the GENEND statement:

```
GENEND UACCTNG=BALANPM  
      UACCTNG=CXJQOSAP
```

See the *NCP, SSP, and EP Resource Definition Reference* for more information about the GENEND definition statement.

SWMMT Definition Statement

SWMMT is a statement valid only for switched subarea lines. The SWMMT statements follow the NCP BUILD and SYSCNTRL statements and are placed before the first GROUP statement.

The format of the SWMMT statement is:

```
name      SWMMT      IDNUM=specific ID
                        NETID=network ID,
                        MAXOUT=number
```

name must match the SWMMTID=*name* on the X25.LINE statement.

For more information about the SWMMT statement, see the *NCP, SSP, and EP Resource Definition Reference*.

X25.CPL Statement (NPSI V3R4 and Later)

Usage: Use the X25.CPL statement to define the CTCP characteristics used with the Fast Transaction Processing Interface (FTPI). You must code one X25.CPL for each FTPI CTCP. All X25.CPL statements must be coded before the first X25.NET statement. Up to 28 X25.CPL statements can be coded.

Position: This statement must be coded before the X25.NET statement.

Format and Keywords: The format of the X25.CPL definition statement is:

```
symbol      X25.CPL      keywords
```

Table 5-1 and Table 5-2 on page 5-15 list the keywords and VTAM keywords that you can code on the X25.CPL statement. The VTAM keywords provide information only to the VTAM initialization process and are not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for descriptions of VTAM keywords and for information on the VTAM initialization process.

Table 5-1 (Page 1 of 2). NPSI Keywords for X25.CPL Statement

Keyword	
ALTCTCP={0,1,...27},	
CTCPNO=number	
[,ADJLINK={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,ADJNODE={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,AGGPRI={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,ANS={CONTINUE CONT}]	
{STOP }	
[,CLRSESSQ={PU,key_value }]	(Starting with V3R9)
{LU,key_value }	
{LU,key_value,PU,key_value }	
[,CMTKYWRD=(targ_name,kywrд_name,kywrд_value)]	(Starting with V3R9)
[,ISTATUS={ACTIVE }]	
{INACTIVE}	
[,LUNAME=name]	
[,MAXTIME={number}]	
{1 }	
[,MINDATA={number}]	
{2 }	
[,MODETAB=table]	
[,NCPGRP={label}]	
{NEW }	
[,NGFINC={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NGFOPT={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NGFTXT={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,PUNAME=name]	
[,SPAN=name]	

Table 5-1 (Page 2 of 2). NPSI Keywords for X25.CPL Statement

Keyword	
[,STATOPT={text string}]	V3R4 - V3R8
[,STATOPT={text string}]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,VTMKYWRD=(targ_name,kywrd_name,kywrd_value)]	(Starting with V3R9)

Table 5-2. VTAM Keywords for X25.CPL Statement

Keyword
[,ASLENT=...]
[,ASLTAB=...]
[,DISCNT=...]
[,DLOGMOD=...]
[,ENCR=...]
[,FEATUR2=...]
[,LOGTAB=...]
[,MDLENT=...]
[,MDLTAB=...]
[,OWNER=...]
[,SSCPFM=...]
[,TERM=...]
[,USSTAB=...]
[,VPACING=...]

symbol

Is an optional label for the name for the X25.CPL statement.

ALTCTCP=(0,1,27)

Specifies a series of up to 27 alternative CTCPs to receive data if the primary CTCP is in a nonresponsive state (for example, if the NOGO command is active). The alternative CTCPs must be coded in the order you want.

The valid value range is from 0–27. Each of the numbers in the series must be unique and different from the CTCPNO keyword specifications on this X25.CPL statement. Each of the numbers in the series must be defined in the CTCPNO keyword of any other X25.CPL statement.

CTCPNO=number

Specifies the individual number of the CTCP assigned by the X25.CPL statement. Each X25.CPL statement has a unique CTCPNO assigned. The valid value range is 0 to 27.

[,ADJLINK={text string}]
 {ALL,text string}
 {GROUP,text string,...LU,text string}

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJLINK keyword and its text string are copied to the LINE statement generated by the X25.CPL statement.

If **ALL**,*text string* is coded, then the ADJLINK keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the ADJLINK keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group.

For example, to code ADJLINK on the PU and LU statements generated by the X25.CPL statement, code

```
ADJLINK=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJLINK=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJLINK=YYYY
```

will be generated following the LU statement.

```
[,ADJNODE={text string    }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJNODE keyword and its text string are copied to the PU statement generated by the X25.CPL statement.

If **ALL**,*text string* is coded, then the ADJNODE keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the ADJNODE keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJNODE on the PU and LU statements generated by the X25.CPL statement, code

```
ADJNODE=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJNODE=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJNODE=YYYY
```

will be generated following the LU statement.

```
[,AGGPRI={text string    }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the AGGPRI keyword and its text string are copied to the LINE statement generated by the X25.CPL statement.

If **ALL**,*text string* is coded, then the AGGPRI keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the AGGPRI keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code AGGPRI on the PU and LU statements generated by the X25.CPL statement, code

```
AGGPRI=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*          AGGPRI=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*          AGGPRI=YYYY
```

will be generated following the LU statement.

```
[,ANS={CONTINUE|CONT}]
      {STOP      }
```

Specifies the ANS keyword that is copied in the NCP PU statement for the CPL resources that are defined. CONTINUE can be abbreviated as CONT.

Code ANS=CONT if you want session continuation for the dependent virtual circuits.

Code ANS=STOP if you do not want session continuation for the dependent virtual circuits.

NPSI provides no default.

```
[,CLRSESSQ={PU,key_value      }]
            {LU,key_value      }
            {LU,key_value,PU,key_value}
```

This keyword and its *key_value* will be generated as a keyword on the generated PU statement or LU statement or both, to be processed by VTAM.

```
[,CMTKYWRD=(targ_name,kywrд_name,kywrд_value)      ]
```

Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrд_name is the name of the keyword to be generated.

kywrд_value is the value to be coded as an operand for the keyword.

```
[,ISTATUS={ACTIVE  }]
            {INACTIVE}
```

Specifies whether the line, physical units, and logical units are activated following a VTAM first start or cold start. This specification is passed to the LINE statement generated by the X25.CPL statement. NPSI provides no default.

```
[,LUNAME=name]
```

Specifies the name of the logical unit associated with the statement. The name assigned can be up to 8 characters.

```
[,MAXTIME={number}]
            {1      }
```

Specifies the maximum time, in tenths of seconds, to wait before sending a PIU to the CTCP. During this time, the received packets are accumulated into a single PIU (unless the PIU count exceeds the maximum number authorized, in which case the PIU is sent as is). The valid value range is 1–20 (.1 to 2 seconds). The default value is 1 (.1).

Note: The exact timer value is not the value coded, but an average value between the coded value and the immediately lower value. For example, if 1 is coded, the average timer value will be $\{(1+0)/2=.5\}$ tenths of a second.

```
[MINDATA={number}]
            {2      }
```

Specifies the *maximum* number of packets to be included in a PIU to be sent to the CTCP. If this number is not reached when the MAXTIME timer elapses, the packet blocking mechanism stops until higher traffic restarts. The valid value range is 1–127. The default value is 2.

X25.CPL

[,MODETAB=*table*]
Specifies the mode table (MODETAB) for the physical circuit LU defined in this X25.CPL statement. The table name can be up to 8 characters.

[NCPGRP={*label*}]
 {NEW }
Is a label applied to the NCP group. If you code NEW, or omit this keyword, the default name for the NCP group will be:

p25CPnnj

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,NGFINC={*text string* }]
 {ALL,*text string*}
 {GROUP,*text string*,...LU,*text string*}

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the STATOPT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.CPL statement.

If **ALL**,*text string* is coded, then the NGFINC keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the NGFINC keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFINC on the PU and LU statements generated by the X25.CPL statement, code

```
NGFINC=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* NGFINC=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* NGFINC=YYYY
```

will be generated following the LU statement.

[,NGFOPT={*text string* }]
 {ALL,*text string*}
 {GROUP,*text string*,...LU,*text string*}

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFOPT keyword and its text string are copied to the PU statement generated by the X25.CPL statement.

If **ALL**,*text string* is coded, then the NGFOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the NGFOPT keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFOPT on the PU and LU statements generated by the X25.CPL statement, code

```
NGFOPT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFOPT=YYYY
```

will be generated following the LU statement.

```
[,NGFTXT={text string      }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFTXT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.CPL statement.

If **ALL**,*text string* is coded, then the NGFTXT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the NGFTXT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFTXT on the PU and LU statements generated by the X25.CPL statement, code

```
NGFTXT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFTXT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFTXT=YYYY
```

will be generated following the LU statement.

```
[,PUNAME=name]
```

Specifies the name of the physical unit associated with the statement. The name assigned can be up to 8 characters.

```
[,SPAN=name]
```

Specifies that the NetView program operator’s access to the CPL resources generated by the X25.CPL statement is restricted. This specification is copied to the LINE, PU, and LU statements generated by the X25.CPL statement.

```
[,STATOPT=text string      ]]
```

V3R4 - V3R8: Specifies whether the NetView Graphics Monitor Facility is used for this resource. The STATOPT keyword and its text string operand are copied to the X25.LINE, X25.PU, and X25.LU statements generated by the X25.CPL statement.

```
[,STATOPT={text string      }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

Starting with V3R9: This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

X25.CPL

If just the *text string* is coded, then the STATOPT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.CPL statement.

If **ALL**, *text string* is coded, then the STATOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.CPL statement.

If the STATOPT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the "targ_name,value" group. For example, to code STATOPT on the PU and LU statements generated by the X25.CPL statement, code

```
STATOPT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* STATOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* STATOPT=YYYY
```

will be generated following the LU statement.

[,VTMKYWRD=(*targ_name*,*kywrд_name*,*kywrд_value*)]

Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrд_name is the name of the keyword to be generated.

kywrд_value is the value to be coded as an operand for the keyword.

X25.END Statement

Usage: Required.

Position: Last in X.25 definition statements.

Format and Keywords: The format of the X25.END definition statement is:
symbol X25.END

Use the X25.END statement to end X.25 generation. Do not include any keywords with this statement.

This definition statement has no keywords.

symbol

Is an optional label for the X25.END statement.

Note: If this statement is not included with your X.25 source statements, NDF will produce a severity 8 warning message. NDF will also flag the following keywords with a severity 4 warning message:

- INCHI
- INCINIT
- INCL2HI
- INCL2LO
- INCPRFX
- LSTUACB
- NCPSTG1
- ORDHI
- ORDINIT
- ORDL2HI
- ORDL2LO
- SRCHI
- SRCLO
- X25VTAM

X25.FCG Statement

Usage: Defines the SNA resources used with fast connect. Use the X25.FCG statement to define SNA resources (LINE, PU, and LU) that will map to SVCs connected to more than one CTCP.

Position: X25.FCG statements follow the corresponding X25.MCH statement in the source statements and follow the X25.VC statements related to that MCH.

The mandatory order of the statements is:

- X25.MCH
- X25.LCG
- X25.VC
- X25.FCG

When you connect several fast connect CTCPs through one physical circuit, you must have one X25.FCG statement for each CTCP. Do *not* code an X25.FCG statement if you use only one CTCP. Code CONNECT=YES on the X25.MCH statement instead. The sample generation in Appendix B, "System Generation Examples," contains sample X25.FCG statements and shows their position within a generation deck.

Format and Keywords: The format of the X25.FCG definition statement is:

```
symbol      X25.FCG      keywords
```

Table 5-3 and Table 5-4 on page 5-23 list the keywords and VTAM keywords that you can code on the X25.FCG statement. The VTAM keywords provide information only to the VTAM initialization process and are not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for descriptions of VTAM keywords and for information on the VTAM initialization process.

Table 5-3 (Page 1 of 2). NPSI Keywords for the X25.FCG Statement

Keyword	
CTCPNO=number, QTY=number	
[,ADJLINK={text string {(ALL,text string) {(GROUP,text string,...LU,text string)}}]	(Starting with V3R9)
[,ADJNODE={text string {(ALL,text string) {(GROUP,text string,...LU,text string)}}]	(Starting with V3R9)
[,AGGPRI={text string {(ALL,text string) {(GROUP,text string,...LU,text string)}}]	(Starting with V3R9)
[,ANS={CONTINUE CONT} {STOP }]	
[,CLRSESSQ={PU,key_value {LU,key_value {LU,key_value,PU,key_value}}	(Starting with V3R9)
[,CMTKYWRD=(targ_name,kywrld_name,kywrld_value)]	(Starting with V3R9)
[,HEXNAME={NO } {YES}]	
[,ISTATUS={ACTIVE } {INACTIVE}]	

Table 5-3 (Page 2 of 2). NPSI Keywords for the X25.FCG Statement

Keyword	
[,MAXDATA={number}]	
{265 }	
[,NCPGRP={NEW }]	
{identifier}	
[,NGFINC={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NGFOPT={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NGFTXT={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,PRFLINE=x]	
[,PRFLU=z]	
[,PRFPU=y]	
[,SPAN=name]	
[,STATOPT={text string }]	(V3R4-V3R8)
[,STATOPT={text string }]	(Starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,SUFFIX=number]	
[,VTMKYWRD=(targ_name,kywrld_name,kywrld_value)]	(Starting with V3R9)

Table 5-4. VTAM Keywords for the X25.FCG Statement

Keyword
[,ASLENT=...]
[,ASLTAB=...]
[,DISCNT=...]
[,DLOGMOD=...]
[,ENCR=...]
[,FEATUR2=...]
[,LOGTAB=...]
[,MDLENT=...]
[,MDLTAB=...]
[,OWNER=...]
[,SSCPFM=...]
[,TERM=...]
[,USSTAB=...]
[,VPACING=...]

symbol

Is an optional name for the X25.FCG statement. It is not used by VTAM or NCP.

CTCPNO=number

Specifies the individual number of the CTCP defined by the X25.FCG statement. See the X25.MCH statement for the CTCP keyword. The range, written in decimal from 0–27, cannot exceed the maximum CTCP number for the physical circuit.

QTY=number

Is the number of simultaneous SNA sessions authorized for the specified CTCP. The number can range from 1–4096, but it must be less than or equal to the number of virtual circuits defined for the physical circuit.

```
[,ADJLINK={text string  }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJLINK keyword and its text string are copied to the LINE statement generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the ADJLINK keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the ADJLINK keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJLINK on the PU and LU statements generated by the X25.FCG statement, code

```
ADJLINK=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJLINK=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJLINK=YYYY
```

will be generated following the LU statement.

```
[,ADJNODE={text string  }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJNODE keyword and its text string are copied to the PU statement generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the ADJNODE keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the ADJNODE keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJNODE on the PU and LU statements generated by the X25.FCG statement, code

```
ADJNODE=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJNODE=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJNODE=YYYY
```

will be generated following the LU statement.

```
[,AGGPRI={text string    }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the AGGPRI keyword and its text string are copied to the LINE statement generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the AGGPRI keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the AGGPRI keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “*targ_name,value*” group. For example, to code AGGPRI on the PU and LU statements generated by the X25.FCG statement, code

```
AGGPRI=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      AGGPRI=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      AGGPRI=YYYY
```

will be generated following the LU statement.

```
[,ANS={CONTINUE|CONT}]
      {STOP      }
```

Specifies the ANS keyword that is copied in the NCP PU statement for the SNA resources that are defined. CONTINUE can be abbreviated as CONT. Code ANS=CONT if session continuation is desired for the dependent virtual circuits. NPSI provides no default.

```
[,CLRSESSQ={PU,key_value    }]
      {LU,key_value    }
      {LU,key_value,PU,key_value}
```

This keyword and its *key_value* will be generated as a keyword on the generated PU statement or LU statement or both, to be processed by VTAM.

```
[,CMTKYWRD=(targ_name,kywrд_name,kywrд_value)    ]
```

Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrд_name is the name of the keyword to be generated.

kywrд_value is the value to be coded as an operand for the keyword.

```
[,HEXNAME={NO  }]
      {YES}
```

HEXNAME is used in conjunction with the PRFLINE, PRFPU, PRFLU, and SUFFIX keywords. When creating resource names by concatenating the PRFLINE, PRFPU, PRFLU names and the SUFFIX values, HEXNAME allows you to convert the SUFFIX value to hexadecimal. HEXNAME is valid only when PRFLINE, PRFPU, and PRFLU are coded. The default value is YES when PRFLINE, PRFPU, and PRFLU are coded.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,ISTATUS={ACTIVE }]
{INACTIVE}

Specifies whether the lines, physical units, and logical units are activated following a VTAM first start or cold restart. NPSI provides no default. This keyword and parameter are copied to the generated NCP LINE statements.

[,MAXDATA={number}]
{265 }

Specifies the maximum number of bytes in a PIU. Because NPSI does not support SNA segmenting for non-SNA DTEs, the value specified for MAXDATA should be greater than the maximum PIU size sent by the application. See the *NCP, SSP, and EP Resource Definition Reference* for calculation and more information about MAXDATA.

[,NCPGRP={NEW }]
{identifier}

Is a label applied to the NCP group. If you code NEW or omit this keyword, the default name for the NCP group will be:

p25PGmnj

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,NGFINC={text string }]
{ALL,text string}
{GROUP,text string,...LU,text string}

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFINC keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the NGFINC keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the NGFINC keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFINC on the PU and LU statements generated by the X25.FCG statement, code

```
NGFINC=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* NGFINC=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* NGFINC=YYYY
```

will be generated following the LU statement.

```
[,NGFOPT={text string  }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFOPT keyword and its text string are copied to the PU statement generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the NGFOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the NGFOPT keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFOPT on the PU and LU statements generated by the X25.FCG statement, code

```
NGFOPT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFOPT=YYYY
```

will be generated following the LU statement.

```
[,NGFTXT={text string  }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFTXT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the NGFTXT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the NGFTXT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFTXT on the PU and LU statements generated by the X25.FCG statement, code

```
NGFTXT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFTXT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFTXT=YYYY
```

will be generated following the LU statement.

```
[,PRFLINE=x]
```

Specifies a user-defined prefix for the name of the LINE statement used for fast connect. The default prefix generated by NPSI is:

pLGmnggg

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,PRFLU=z]

Specifies a user-defined prefix for the name of the LU used for fast connect. The default prefix generated by NPSI is:

pUGmnggg

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,PRFPU=y]

Specifies a user-defined prefix for the name of the PU used for fast connect. The default prefix generated by NPSI is:

pPGmnggg

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,SPAN=name]

Specifies that the NetView program operator’s access to the virtual circuit line is restricted. This specification is copied to the LINE, PU, and LU statements generated by the X25.FCG statement.

[,STATOPT=*text string*]

V3R4 - V3R8: Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.

[,STATOPT={*text string* }]

{**ALL**,*text string*}

{**GROUP**,*text string*,...**LU**,*text string*}

Starting with V3R9: This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the STATOPT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.FCG statement.

If **ALL**,*text string* is coded, then the STATOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.FCG statement.

If the STATOPT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code STATOPT on the PU and LU statements generated by the X25.FCG statement, code

```
STATOPT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* STATOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* STATOPT=YYYY
```

will be generated following the LU statement.

[,SUFFIX=*number*]

Specifies a number that is the lower limit for each set of names built. There is no default. This keyword must be coded together with PRFLINE, PRFPU, and PRFLU if user-defined names are to be generated. The length of the names is equal to the length of PRFLINE, or PRFPU, or PRFLU, added to the length of the SUFFIX parameter. Values range from 0–X'FFF' if HEXNAME=YES, and from 0–4095 if HEXNAME=NO.

PRFLINE, PRFPU, PRFLU, and SUFFIX must be coded together.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names and how to code the related statement keywords.

[,VTMKYWRD=(*targ_name*,*kywrd_name*,*kywrd_value*)]

Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrd_name is the name of the keyword to be generated.

kywrd_value is the value to be coded as an operand for the keyword.

X25.LCG Statement

Usage: Use this statement to describe the LCG. You must code an X25.LCG statement for each LCG defined on a physical circuit.

Position: This statement follows the X25.MCH statement and precedes the virtual circuit definition for this group.

Format and Keywords: The format of the X25.LCG statement is:
 X25.LCG keywords

Table 5-5 shows the keywords you can code on the X25.LCG statement.

Table 5-5. X.25 Keywords for X25.LCG Statement

Keyword

LCGN=*number*

LCGN=*number*

Specifies the logical channel group number (LCGN) for all virtual circuits specified by the subsequent X25.VC or X25.LINE statements. Valid values range from 0–15. You must define the number in the LCGDEF keyword of the preceding X25.MCH statement.

Note: If you use more than one LCG, code the LCG numbers in the X25.LCG statements in ascending order.

X25.LINE Statement

Usage: Use the X25.LINE statement to define the characteristics of a virtual circuit. This statement is similar to the X25.VC statement. The X25.LINE statement generates an NCP LINE statement and allows you to control the LINE name and some parameters of the NCP LINE statement.

Position: Arrange X25.LINE statements that define virtual circuits associated with the same logical channel group in ascending order by logical channel number (LCN). Each X25.LINE statement must be followed by one X25.PU statement. If you are defining a non-subarea node PVC, it also must be followed by at least one X25.LU statement.

X.25 subarea links must be defined with an X25.LINE statement followed by an X25.PU statement. For PVCs, the LCN is defined in the X25.LINE statement. For SVCs, the LCN is defined in an X25.VC statement. This means that each SVC subarea connection needs one X25.LINE, and one X25.PU definition. The X25.VC definition statement can contain more than one subarea virtual circuit, as well as boundary network node (BNN) virtual circuits.

Format and Keywords: The format of the X25.LINE statement is:
 symbol X25.LINE keywords

Table 5-6 on page 5-32 and Table 5-7 on page 5-33 lists the keywords and VTAM keywords that you can code on the X25.LINE statement. The VTAM keywords provide information only to the VTAM initialization process and is not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for a description of VTAM keywords and for information about the VTAM initialization process.

You cannot modify the X.25 level 3 DTE timer with X25.LINE. These values, set according to the CCITT 84 Recommendations, are as follows:

Timer and Seconds	Packet
T20=180 seconds	Restart Request
T21=200 seconds	Call Request
T22=180 seconds	Reset Request
T23=180 seconds	Clear Request
T25=0 seconds	Window Rotation Timer
T26=180 seconds	Interrupt Confirmation. Specifies the waiting time for Interrupt Confirmation under integrated PAD or GATE. The valid range is from 10–255; the default is 180 seconds.

If these values do not meet your specific requirements, use the DATE function to select appropriate values or select the values using the T20, T21, T22, T23, T25, or T26 keywords of the X25.NET statement.

Table 5-6 (Page 1 of 2). X.25 Keywords for X25.LINE Statement

Keywords	
TYPE={SWITCHED S }, {PERMANENT P}, [,ADJLINK(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,ADJNODE(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,AGGPRI(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,CALL={OUT }] {IN } {INOUT}	
[,CMTKYWRD=(targ_name,kywrd_name,kywrd_value)]	(X.25 ODLC and NPSI starting with V3R9)
[,COMMITO={1}] {2} {4} {8}	
[,CTCPNO=number]	(V3R4 and Later)
[,DEFPRI={NO }] {LOW } {NORMAL} {HIGH1 } {HIGH2 }	(V3R4 and Later)
[,DSTNODE={BNN}] {INN}	
[,ID={bbbfffff}]	
[,IDNUMT={ffff}]	IDNUMT must be an even number
[,IPL={NO }] {YES}	
[,ISTATUS={ACTIVE }] {INACTIVE}	
[,LCN=number]	
[,LLC={LLC0}] {LLC2} {LLC3} {LLC4} {LLC5}	
[,MODE={SECONDARY}] {SEC }	
[,MONLINK={YES }] {NO } {CONTINUOUS} {CONT }	
[,NCPGRP={label}] {NEW }	
[,NGFINC=(text string)]	(X.25 ODLC and NPSI starting with V3R8)
[,NGFOPT=(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,NGFTEXT=(text string)]	(X.25 ODLC and NPSI starting with V3R8)
[,NOTOPC5={YES}] {NO }	(V3R4 and Later)
[,OSITYPE={number}] {0 }	
[,OUFINDX=index]	
[,RESETUNB={YES}] {NO }	(Starting with V3R9)

Table 5-6 (Page 2 of 2). X.25 Keywords for X25.LINE Statement

Keywords

[,RETVCCCT={count}]	
{3 }	
[,RETVCTO={timeout}]	
{30 }	
[,SPAN=name]	
[,STATOPT=text string]	
[,SWMTID=name]	
[,TADDR=chars]	
[,TRANSFR=count]	
[,VCCINDX=index]	
[,VTMKYWRD=(targ_name,kywrд_name,kywrд_value)]	(X.25 ODLC and NPSI starting with V3R9)

Table 5-7. VTAM Keywords for X25.LINE Statement

VTAM Keywords

[,DYNPU=...]
 [,DYNPUFX=...]
 [,OWNER=...]

symbol

Is an optional name that labels the corresponding LINE statement in the NCP source statements, and serves as the symbolic name for the virtual circuit that you define. The default name generated by NPSI is:

pLxxxggg

See “Resource Naming Conventions” on page 5-2 for information on default naming conventions.

**TYPE={SWITCHED|S }
 {PERMANENT|P }**

Specifies the virtual circuit type. This keyword is required and has no default value. To define an SVC, code TYPE=SWITCHED or TYPE=S. To select a PVC, code TYPE=PERMANENT or TYPE=P. If the MCH is dedicated to fast connect, code TYPE=S.

[,ADJLINK=*text string*]

This keyword will be generated as a comment following the NCP LINE statement, to be processed by NetView.

[,ADJNODE=*text string*]

This keyword will be generated as a comment following the NCP LINE statement, to be processed by NetView.

[,AGGPRI=*text string*]

This keyword will be generated as a comment following the NCP LINE statement, to be processed by NetView.

[,CALL={OUT }]
 {IN }
 {INOUT }

Specifies the way an SVC is established. You can use this keyword only if TYPE=SWITCHED or TYPE=S is coded.

Use CALL=IN if the virtual circuit that you define will be established at the request of a remote DTE.

Do not specify CALL=IN on any virtual circuit definition following an X25.MCH statement that specifies a value greater than 0 for the SVCINN keyword.

X25.LINE

Use CALL=OUT if the virtual circuit that you define will be established at the request of the local host access method.

CALL=INOUT means that a virtual circuit can be established in either way.

For a DATE physical circuit, you must code the CALL keyword to match your PSDN subscription.

For a GATE MCH, if you want to reserve a certain number of virtual circuits for non-GATE call-outs, code CALL=OUT for the required number of virtual circuits. The number of virtual circuits used to call out by the CTCP should be declared as CALL=INOUT or CALL=IN.

For subarea lines, the default value is INOUT.

Note: CALL=OUT must not be coded on an X25.LINE statement belonging to an MCH dedicated to FTPI.

[,CMTKYWRD=(*targ_name*,*kywrд_name*,*kywrд_value*)]

Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrд_name is the name of the keyword to be generated.

kywrд_value is the value to be coded as an operand for the keyword.

[,COMMITO={1}]

{2}

{4}

{8}

Specifies, in seconds, the idle virtual circuit commit timeout. When this time elapses, buffers that were committed for this virtual circuit are decommitted. The default value is 4.

Do not specify COMMITO when TYPE=SWITCHED or TYPE=S, and DSTNODE=INN.

NCP X.25 ODLC ignores the value specified for the COMMITO keyword.

[,CTCPNO=*number*]

V3R4 and Later: Specifies, in decimal, the individual number assigned to the FTPI CTCP with which this virtual circuit communicates. This number is the CTCPNO value coded on the X25.CPL statement of the selected CTCP. This parameter applies only to PVCs belonging to an FTPI MCH, and must be used when there are several FTPI CTCPs. You can omit CTCPNO when only one X25.CPL is coded. You must not code CTCPNO if the X25.CPL statement is not coded (that is, if FTPI is not selected).

[,DEFPRI={NO }]

{LOW }

{NORMAL}

{HIGH1 }

{HIGH2 }

V3R4 and Later: Specifies the QOS priority value for the GATE PVC.

Note: If you code DEFPRI=LOW|NORMAL|HIGH1|HIGH2 on the X25.MCH statement, you cannot code DEFPRI=NO on the X25.LINE statement. If you code DEFPRI=NO on the X25.MCH statement (or if DEFPRI is not coded on the MCH), you cannot code DEFPRI=LOW|NORMAL|HIGH1|HIGH2 on the X25.LINE statement. It is valid on the X25.LINE statement only when

TYPE=P, and LLC=4 are coded. The default value is the value of the DEFPRI keyword coded on the X25.MCH statement.

[,DSTNODE={BNN}]
{INN}

Specifies the type of destination node. The DSTNODE keyword is meaningful only for LLC type 3. DSTNODE=BNN means that the virtual circuit is connected to a peripheral node, such as a cluster or a terminal. DSTNODE=INN means that the virtual circuit is connected to a subarea node, such as another communication controller. The default value is BNN. You can specify DSTNODE=INN together with TYPE=SWITCHED or TYPE=S when SVCINN is coded with a nonzero value on the X.25 MCH statement. If DSTNODE=INN and TYPE=S are coded, SVCINN must be coded on the X25.MCH statement.

[,ID={bbbfffff}]

Specifies the ID parameter used to support the NCP anonymous caller function, where *bbb* is the IDBLK value and *fffff* represents five hexadecimal digits and is the IDNUM value on the X.25 LINE group. If ID is not specified, the value is taken from the IDBLK/IDNUM of the remote DTE.

Notes:

1. The ID keyword can only be applied to LLC type 2, LLC type 3 without SVCSC, and LLC type 0 without subaddressing or DATE.
2. You must either code the ID keyword on all SNA and PCNE SVCs of an MCH, or do not code the ID keyword on any of them.

[,IDNUMT={ffff}]

Specifies four hexadecimal digits to add to the hexadecimal digit specified by the NCP BUILD X25.IDNUMH keyword to form the IDNUM that NPSI can use for connections with a non-SNA DTE. See Table 4-1 on page 4-12 for the circumstances that use IDNUM to establish virtual calls. The default is a number that is generated internally by NPSI based on the order of the SVCs in the generation. The IDNUMT keyword must be an even number.

Notes:

1. IDNUMT cannot be coded on an X25.LINE statement belonging to a fast connect or FTPI MCH.
2. You must either code the IDNUMT keyword on all non-SNA SVCs for all the MCHs of the entire NPSI generation, or not code the IDNUMT keyword on any of them.

[,IPL={YES}]
{NO}

Specifies whether the NCP being generated can be loaded and dumped over this line. The default is NO.

[,ISTATUS={ACTIVE }]
{INACTIVE}

Specifies whether the line is activated following a VTAM first start or cold restart. X.25 does not provide a default. This keyword and parameter are reported on the generated NCP LINE statements.

[,LCN=*number*]

Specifies the LCN of the defined virtual circuit. Code this value in decimal notation. Valid values range from 0–255.

Do not specify LCN when TYPE=SWITCHED or TYPE=S, and DSTNODE=INN. For all other combinations of TYPE and DSTNODE, specified or defaulted to, LCN must be specified.

X25.LINE

For X.25 ODLC, logical channels must be defined according to CCITT (in other words, in four groups of consecutively numbered LCNs). These groups must be in the following order:

1. PVCs
2. SVCs with CALL=IN
3. SVCs with CALL=INOUT
4. SVCs with CALL=OUT

There may be gaps in LCNs between groups.

Note: Some networks do not use the value 0 within logical channel group 0.

```
[,LLC={LLC0}]  
    {LLC2}  
    {LLC3}  
    {LLC4}  
    {LLC5}
```

Specifies the type of PVC defined by this statement. This keyword is used only for PVCs, because the LLC type for SVCs is defined at call setup. This keyword is mandatory when you code TYPE=PERMANENT or TYPE=P.

- Code LLC0 if the virtual circuit goes to a non-SNA DTE and does not require GATE or PAD support (type 0 virtual circuit).
- Code LLC2 if the virtual circuit goes to an SNA peripheral node (type 2 virtual circuit) attached with PSH protocol.
- Code LLC3 if the virtual circuit goes to an SNA peripheral node attached with QLLC protocol or to another subarea node (type 3 virtual circuit).
- Code LLC4 if the virtual circuit requires GATE support (type 4 virtual circuit).
- Code LLC5 if the virtual circuit connects to a terminal requiring PAD support (type 5 virtual circuit). Specify the type of PAD support (integrated or transparent) using the PAD keyword of the X25.MCH statement.

```
[,MODE={SECONDARY}]  
    {SEC    }
```

The MODE keyword is applicable only to casual connection on PVC. MODE specifies the type of operation for a line group. MODE=SEC must be coded on the secondary side. You must not code the primary side. The NDF facility generates one GROUP statement when a change is detected in the MODE keyword. It is not recommended to interleave primary and secondary mode definitions. MODE is valid only if LLC3 is coded on the X25.LINE statement. If MODE=SEC, you must code the TADDR keyword in X25.LINE.

```
[,MONLINK={YES    }]  
    {NO        }  
    {CONTINUOUS}  
    {CONT      }
```

Specifies the value given to this keyword in the corresponding NCP LINE statement. There is no default value. Code YES if this is a remote NCP to be activated over the subarea node PVC. See the *NCP, SSP, and EP Resource Definition Reference* for details.

If the only connection to the communication controller is over a PSDN, the associated X.25 MCH statement must specify XMONLNK=YES and ANS=CONT if any subarea communication PVC specifies MONLINK=YES.

[,NCPGRP={*label*}]
 {**NEW** }

Specifies the label of the NCP GROUP statement that includes the virtual circuit. If you do not specify this value, the name is generated by NDF. You can code this keyword if you want to modify the group name to which this LINE belongs.

If you specify a *label*, a new NCP GROUP statement is created that uses this label as its name.

If you specify NEW, a new NCP GROUP statement is automatically created at X.25 generation.

V3R3 Only: For V3R3, NDF creates one group of PVCs, one group of SVCs, and one group of SVCSCs by default for each X25.MCH definition.

X.25 ODLC and NPSI V3R4 and Later: NDF creates four groups by default for each X25.MCH definition. One group is created for INN PVCs, another for BNN PVCs, a third for INN SVCs, and a fourth for BNN SVCs.

If you omit the NCPGRP keyword, the virtual circuit will belong to the group with the most recently created name. The default name is generated as follows:

p25Pxxxy for PVCs
 p25Sxxxy for SVCs

See "Resource Naming Conventions" on page 5-2 for information on default naming conventions.

Note: This keyword can no longer be used to make a call-out on a virtual circuit belonging to a given set of virtual circuits within an MCH. If a virtual circuit is within an MCH, X.25 allocates the virtual circuits for call-outs starting from the largest virtual circuit number available for call-out.

[,NGFINC=*text string*]

X.25 ODLC and NPSI starting with V3R8: Specifies whether the NetView status monitor is used for this resource. The NGFINC keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.

[,NGFOPT=*text string*]

X.25 ODLC and NPSI starting with V3R9: Specifies whether the NetView status monitor is used for this resource. The NGFINC keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.

[,NGFTEXT=*text string*]

X.25 ODLC and NPSI starting with V3R8: Specifies whether the NetView status monitor is used for this resource. The NGFTEXT keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.

[,NOTOPC5={**YES**}]
 {**NO** }

V3R4 and Later: The NOTOPC5 keyword is applicable only on a PVC defined as LLC=LLC3 and DSTNODE=BNN. Valid values are YES and NO. The default value is NO. Specify NOTOPC5=YES when a PVC connecting a DTE is reset with diagnostic X'C5' when the DTE becomes non-operational and with diagnostic X'A1' when the DTE becomes operational again.

Only NO can be coded for X.25 ODLC.

X25.LINE

[,OSITYPE={number}]
 {0 }

Specifies, in decimal format, the GATE CTCP with which this virtual circuit must communicate. This parameter applies to PVCs only, and must be used when the MCH is capable of communicating with several GATE CTCPs. If the MCH is not capable of communicating with more than one GATE CTCP, this keyword must be omitted. Valid values range from 0–27. The default value is 0.

Notes:

1. A PVC can be bound to any CTCP, regardless of the CTCP number and the OSITYPE. If a control packet (for example, Reset) received from the network is to be sent over the control session, NPSI uses the OSITYPE keyword to direct the packet to an appropriate CTCP.
2. A GATE MCH can communicate with more than one CTCP when it has SVCs that are controlled by several CTCPs (see the CTCP, CUD0, and SUBD keywords on the X25.MCH statement). However, when an MCH has only PVCs, or PVCs and SVCs controlled by the same CTCP, there can be only one CTCP. When this situation occurs, the OSITYPE keyword must be either omitted, or coded as OSITYPE=0 on the X25.LINE statement.

[,OUFINDX=index]

Specifies the default index value in the X25.OUFT table. The index value points to the table entry that builds the facility field and the end of the CUD field in a Call Request packet. OUFINDX serves as a default for call-outs if you code yy=00 in the DIALNO or DLCADDR keywords on the VTAM switched major node PATH statement. Valid values range from 1–99. This keyword is required when you code TYPE=SWITCHED or TYPE=S. This keyword is not required when you code DSTNODE=INN.

Note: Do not code OUFINDX on an X25.LINE statement belonging to an FTPI MCH.

[,RESETUNB={YES}]
 {NO }

Starting with V3R9: Specifies whether NPSI should send a RESET packet when a BIND or UNBIND is received from the application, or a data packet is received but the PCNE LU session is not bound. NO is the default.

When RESETUNB=YES is coded, MAXOUT=6 must be coded on the X25.VC or X25.PU statement that defines the PCNE PVC.

RESETUNB=YES is allowed only for LLC0 PVCs. RESETUNB=YES is not allowed when GATE=DEDICAT is specified on the X25.MCH statement.

[,RETVCT={count}]
 {3 }

Specifies the number of retransmissions of a physical services command. RETVCT is used only when the virtual circuit goes to an SNA DTE. Valid values range from 0–255; the default value is 3.

[,RETVCT0={timeout}]
 {30 }

Specifies the time between retransmissions of physical services commands when the virtual circuit is connected to an SNA DTE. Valid values range from 0–255 seconds; the default value is 30 seconds.

For the IBM 3710, specify a value greater than the time needed to perform recovery for a downstream device.

- [,SPAN=*name*] Specifies that the NetView program operator's access to the virtual circuit line is restricted. This specification is copied to the LINE, PU, and LU statements generated by the X25.LINE statement.
- [,STATOPT=*text string*] Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.
- [,SMMTID=*name*] This keyword is valid only on a LINE dedicated to SVCSC. It contains the name of a switched SMMF table defined by a group of SWMMT statements.
- [,TADDR=*chars*] The TADDR keyword is required if you specify MODE=SEC. TADDR specifies a symbolic link station address for a type 4 PU NCP on a subarea link.
- chars* is the 2-digit hexadecimal representation of the EBCDIC character you choose for the symbolic address. If you omit TADDR, the default is the EBCDIC representation of the subarea address of this NCP. For example, if this NCP has a subarea address of 3 and a symbolic address of D, code TADDR=C4. However, if you do not code TADDR, the default address for this controller is generated as TADDR=03.
- [,TRANSFR=*count*] Specifies the number of NCP buffers corresponding to the maximum amount of data that can be received from the line during a single data transfer operation. This keyword applies only to virtual circuits working in LLC type 3.
- Valid values range from 1–254. The count must be equal to or less than the TRANSFR count coded or defaulted in the BUILD statement.
- If the virtual circuit connects an SNA PU type 2.1, the checking of inbound data against the X25.MAXPIU value on the BUILD statement is replaced by the checking of the value of TRANSFR times the NCP buffer size (BFRS on the BUILD statement) minus the offset to the received data in the first buffer (38).
- [,VCCINDX=*index*] Specifies the entry in the connection parameter table that contains the operational characteristics of the virtual circuit defined by this X25.LINE statement.
- Valid values range from 1–99. The index serves as a default for call-outs if you code XX=00 in the DIALNO or DLCADDR keywords.
- Packet and window sizes defined in the corresponding entry of the VCCPT must be the same as the values assigned by the PSDN.
- Do not specify VCCINDX when TYPE=SWITCHED or TYPE=S, and DSTNODE=INN. For all other combinations of TYPE and DSTNODE specified or defaulted to, VCCINDX must be specified.
- When VCCINDX is coded for SVCs, the value must be the same on all of the X25.LINE statements for this MCH.
- If VCCINDX is specified on the X25.MCH statement, the value of VCCINDX from the MCH statement will be used if not coded here.
- If this is a PVC and VCCINDX is coded on both the X25.MCH and X25.LINE statement, the value coded on the X25.LINE statement will override the value specified on the X25.MCH statement.
- If this is an SVC, the value must match what is coded on the X25.MCH statement if it is coded here.

X25.LINE

[,VTMKYWRD=(*targ_name*,*kywrд_name*,*kywrд_value*)]

Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrд_name is the name of the keyword to be generated.

kywrд_value is the value to be coded as an operand for the keyword.

X25.LU Statement

Usage: Use this statement to define an LU associated with the DTE. The keywords are the same as those for the NCP LU statement.

Position: This statement follows the X25.PU statement for PVC definitions.

Format and Keywords: The format of the X25.LU statement is:
 symbol X25.LU keywords

Table 5-8 on page 5-42 and Table 5-9 on page 5-42 list the keywords and VTAM keywords that you can code on the X25.LU statement. The VTAM keywords provide information only to the VTAM initialization process and are not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for descriptions of VTAM keywords and for information about the VTAM initialization process.

Code one X25.LU statement for each logical unit associated with a virtual circuit. For more information about the LU statement and its keywords, see the *NCP, SSP, and EP Resource Definition Reference*.

X25.LU is not allowed under an X25.LINE–X25.PU pair defining an FTPI PVC.

Multiple X25.LUs are not allowed for resources connected by non-SNA virtual circuits (PCNE, PAD, GATE).

For Start/Stop Terminal Users: Depending on your application, when a PVC is used to communicate with a Start/Stop terminal (TWX device), you must consider the following information for GROUP and LU macros.

If the Start/Stop terminal is using a PVC link, you might want to code the following in the NEWDEFN supplied to VTAM in SYS1.VTAMLST:

SSCPFM=USSNTO

Code SSCPFM=USSNTO in the LU macro to specify that the simulated LU uses character-coded RUs (with the carriage return as the delimiter) in the SSCP-LU session.

TERM=TWX

Code TERM=TWX in the LU macro to ensure that a carriage return and a linefeed character are sent at the end of each output RU from applications such as TSO.

VIRTUAL=YES

Code VIRTUAL=YES in the GROUP macro to define the group of PVCs that are supported by the LLC type 5 (PAD) function. If this value is not coded, the TERM=TWX operand is ignored, and you have either a carriage return or linefeed problem, or both, with your output data stream.

You must add the VIRTUAL=YES operand to the NEWDEFN supplied to VTAM in VTAMLST *after* the NPSI/NCP generation is complete. If the VIRTUAL=YES operand is specified in the GROUP macro before NCP system generation, the assembly indicates that you must specify LINEFVT, PUFVT, and LUFVT. This is not valid for the NPSI/NCP environment. Including the VIRTUAL=YES operand after NPSI/NCP generation provides effective use of this operand as a VTAM-only macro, and directs VTAM not to reject session requests for this TWX device. If

VIRTUAL=YES is not specified, the session is rejected by the SSCP when session initiation begins.

Table 5-8. X.25 Keywords for X25.LU Statement

Keywords	
[ADJLINK(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,ADJNODE(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,AGGPRI(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,CMTKYWRD=(targ_name,kywrд_name,kywrд_value)]	(X.25 ODLC and NPSI starting with V3R9)
[,LOCADDR=number]	
[,LUDR=NO YES]	
[,NGFINC(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,NGFOPT(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,NGFTXT(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,STATOPT=text string]	
[,VTMKYWRD=(targ_name,kywrд_name,kywrд_value)]	(X.25 ODLC and NPSI starting with V3R9)

Table 5-9. VTAM Keywords for X25.LU Statement

Keyword	
[,ASLENT=...]	
[,ASLTAB=...]	
[,BATCH=...]	
[,DLOGMOD=...]	
[,ENCR=...]	
[,FEATUR2=...]	
[,ISTATUS=...]	
[,LOGAPPL=...]	
[,LOGTAB=...]	
[,MAXSESS=...]	(V3R6 and Later)
[,MDLENT=...]	
[,MDLTAB=...]	
[,MODETAB=...]	
[,NETID=...]	(V3R6 and Later)
[,NPACOLL=...]	
[,PACING=...]	
[,REGISTER=...]	(V3R6 and Later)
[,RESSCB=...]	
[,SPAN=...]	
[,SSCPFM=...]	
[,TERM=...]	
[,USSTAB=...]	
[,VPACING=...]	

symbol

Is an optional field that provides a resource name for the logical unit. The default name is:

pUxxxxzz

When multiple LUs are attached to the same PU, you must use *symbol* on each X25.LU statement. However, SNA naming conventions do not permit the term *symbol* to be duplicated within a single network.

See “Resource Naming Conventions” on page 5-2 for information on default naming conventions.

- [,ADJLINK=*text string*] This keyword will be generated as a comment following the NCP LU statement, to be processed by NetView.
- [,ADJNODE=*text string*] This keyword will be generated as a comment following the NCP LU statement, to be processed by NetView.
- [,AGGPRI=*text string*] This keyword will be generated as a comment following the NCP LU statement, to be processed by NetView.
- [,CMTKYWRD=(*targ_name,kywrd_name,kywrd_value*)]
 Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement.
targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.
kywrd_name is the name of the keyword to be generated.
kywrd_value is the value to be coded as an operand for the keyword.
- [,LOCADDR=*number*] Specifies the local address of the logical unit in decimal with no leading zeros. For LLC types 0, 4, and 5 you must code LOCADDR=0.
- [,LUDR=NO|YES] **V3R6 and Earlier:** Specifies whether the logical unit can be deleted by dynamic reconfiguration. Because NPSI does not support dynamic reconfiguration, NPSI allows LUDR to be coded as NO or to be omitted.
X.25 ODLC and NPSI V3R7 and later: You can code LUDR=YES under the following conditions:
 - TYPE=PERMANENT is defined on the corresponding X25.LINE statement.
 - LLC=LLC3 is defined on the corresponding X25.LINE statement.
- [,NGFINC=*text string*] Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LU statement generated by NDF.
- [,NGFOPT=*text string*] Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LU statement generated by NDF.
- [,NGFTXT=*text string*] Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LU statement generated by NDF.
- [,STATOPT=*text string*] Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LU statement generated by NDF.
- [,VTMKYWRD=(*targ_name,kywrd_name,kywrd_value*)]
 Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.
targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrd_name is the name of the keyword to be generated.

kywrd_value is the value to be coded as an operand for the keyword.

X25.MCH Statement

Usage: Use this statement to describe an X.25 physical circuit. It generates an NCP LINE statement, an NCP PU statement, and one or more NCP LU statements.

Position: This statement follows the X25.NET statement. The X25.MCH statement is then followed by the X25.LCG statement and other statements that define the virtual circuits.

Notes:

1. For Fast Connect, a maximum of 36 X25.MCHs can be coded with CONNECT=CUD0 or CONNECT=SUBD. In addition, if more than 36 X25.MCHs are coded, (both Fast Connect and non-fast connect), the fast connect MCHs must be coded among the first 36 X25.MCHs.
2. X.25 ODLC and NPSI operates in a non-NRZI mode. The terminal equipment must also be configured in non-NRZI mode.

Format and Keywords: The format of the X25.MCH statement is:

```
symbol      X25.MCH      keywords
```

Table 5-10 and Table 5-11 on page 5-49 list the keywords and VTAM keywords that you can code on the X25.MCH statement. The VTAM keywords provide information only to the VTAM initialization process and are not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for descriptions of VTAM keywords and for information about the VTAM initialization process.

Table 5-10 (Page 1 of 5). X.25 Keywords for X25.MCH Statement

Keyword	
ADDRESS={xxxx}, FRMLGTH=length, LCGDEF=(lcg,lcghi), or (lcg,lcghi),(lcg,lcghi),...), MWINDOW=window	
[,ACCOUNT={YES {NO {(LLC0,LLC2,LLC3I,LLC3B,LLC4,LLC5)}}]	(X.25 ODLC and NPSI V3R4 and later)
[,ACTIVTO=timer] [,ADDIFG={NO }] {YES}	(X.25 ODLC and NPSI V3R8 and Later)
[,ADJLINK={text string {(ALL,text string) {(GROUP,text string,...LU,text string)}}]	(X.25 ODLC and NPSI starting with V3R9)
[,ADJNODE={text string {(ALL,text string) {(GROUP,text string,...LU,text string)}}]	(X.25 ODLC and NPSI starting with V3R9)
[,AEFBYTE=(aa,bb,...)] [,AEFCTCP=(ii,jj,...)]	
[,AGGPRI={text string {(ALL,text string) {(GROUP,text string,...LU,text string)}}]	(X.25 ODLC and NPSI starting with V3R9)
[,ANS={CONTINUE CONT] {STOP }]	
[,BRKCON={CONNECTO} {NOWNERTO} {NONE }]	

Table 5-10 (Page 2 of 5). X.25 Keywords for X25.MCH Statement

Keyword	
[,CAUSE=DCE]	
[,CCXDELAY={timer}]	
{NONE }	
{2 }	
[,CLRSESSQ={PU,key_value }]	(X.25 ODLC and NPSI starting with V3R9)
{LU,key_value }	
{LU,key_value,PU,key_value }	
[,CMTKYWRD=(targ_name,kywr_name,kywr_value)]	(X.25 ODLC and NPSI starting with V3R9)
[,CONNECT={NO }]	
{YES }	
{CUD0 }	
{SUBD }	
[,CTCP=(m1,m2,m3,...,mn)]	
[,CUD0={({n1,n2,n3,...,nn,NULL})}]	
(n1,n2,n3,...,nn,ALL)	(Starting with V3R9)
{ALL }	
[,DBIT={YES}]	
{NO }	
[,DEFPRI={NO }]	(V3R4 and later)
{LOW }	
{NORMAL }	
{HIGH1 }	
{HIGH2 }	
[,DIRECT={YES}]	
{NO }	
[,DSABLTO=timer]	
[,DTETODTE={NO }]	(X.25 ODLC)
{YES }	
[,ENABLTO=timer]	
[,FTPI={NO }]	(V3R4 and later)
{YES }	
{CUD0 }	
{SUBD }	
{PVC }	(V3R8 and later)
[,FTPITH=threshold]	(Starting with V3R9)
[,FTPITHTO=timeout]	(Starting with V3R9)
[,GATE={DEDICAT}]	
{GENERAL }	
{NO }	
[,GATEINFO={NO }]	(Starting with V3R9)
{YES }	
[,IDBLKC={ccc}]	
[,IDBLKG={ggg}]	
[,IDBLKLG={({lcg,idblock})}]	(V3R8 and later)
{((lcg,idblock),(lcg,idblock),...)}]	
[,IDBLKP={ppp}]	
[,INTFAC={NO }]	
{YES }	
[,INTWLINK={NO }]	(V3R6 and later)
{DIALDTE }	
{DIALPSDN }	
[,ISTATUS={ACTIVE }]	
{INACTIVE }	
[,ITRACE={YES}]	
{NO }	
[,LCN0={NOTUSED}]	
{USED }	

Table 5-10 (Page 3 of 5). X.25 Keywords for X25.MCH Statement

Keyword	
[,LLCLIST=(LLC0,LLC2,LLC3,LLC4,LLC5)]	
[,{LLC0=(m1,m2...),...}]	
{LLC2=(m1,m2...)} {LLC3=(m1,m2...)} {LLC4=(m1,m2...)} {LLC5=(m1,m2...)} {LLCI=(m1,m2...)} [,LOGAPPL=(app10,app11,...,app1m)]	
[,LSPRI={YES}]	
{NO }	
[,LUNAME=(luname0,luname1,...lunamem)]	
[,MACB=(labelx,labelr)]	
[,MBITCHN={YES}]	
{NO }	
[,MBNOFULL={YES}]	
(ACC)	
{NO }	
[,MMODULO={8 }]	(X.25 ODLC and NPSI V3R7 and later)
{128}	
[,MODETAB=(table0,table1,...tablem)]	
[,NCPGRP={label}]	
{NEW }	
[,NDRETRY={count}]	
{1 }	
[,NETID=(name,ANY)]	
[,NGFINC={text string }]	(X.25 ODLC and NPSI starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NGFOPT={text string }]	(X.25 ODLC and NPSI starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NGFTXT={text string }]	(X.25 ODLC and NPSI starting with V3R9)
{(ALL,text string)}	
{(GROUP,text string,...LU,text string)}	
[,NPACOLL={MCHLINE,MCHPU,VCPU}]	(X.25 ODLC and NPSI V3R4 and later)
{NO,NO,NO }	
[,NPADTEAD=number]	(X.25 ODLC and NPSI V3R4 and later)
[,NPPVCN=number]	(V3R4 and later)
[,NPRETRY={count}]	
{7 }	
[,ODLC={NO }]	(X.25 ODLC)
{YES }	
[,OWNER={owner_name }]	(X.25 ODLC and NPSI V3R4 and later)
{owner_name1,owner_name2,...}	
{owner_name1,PASS }	
{owner_name1,PASSTHRU }	
[,PAD={INTEG }]	
{TRANSP}	
{NO }	
[,PADBRKCD={{(cc1dd1,cc2dd2,...)}}]	
{0084,0085 }	
[,PADCHECK={NO }]	
{YES}	
[,PADINDX={index}]	
{0 }	
[,PHYSRSC={YES}]	(X.25 ODLC and NPSI V3R5 and later)
{NO }	

Table 5-10 (Page 4 of 5). X.25 Keywords for X25.MCH Statement

Keyword	
[,PKTMODL={128}]	
{8 }	
[,PLPIGGYB={YES}]	
{NO }	
[,PUNAME=puname]	
[,PWPROT={YES }]	
{YESWOCC}	
{NO }	
[,RESETPVC={YES}]	(V3R7 and later)
{NO }	
[,RESETUNB={YES}]	(Starting with V3R9)
{NO }	
[,RNRPKT={YES}]	
{NO }	
[,RNRTIMER={nnn}]	(V3R7 and later)
{0 }	
[,ROLE={PRI}]	
{NEG}	
[,RPADPARM={YES}]	
{NO }	
[,SDRTCNT=count]	
[,SDRTIME=time]	
[,SHM={YES}]	
{NO }	
[,SHUTD={INVCLR }]	
{NOINVCLR}	
[,SPAN=name]	
[,SPEED={rate }]	
{9600 }	
{245760 }	(V3R5 and later)
{256000 }	(V3R5 and later)
[,SPNQLLC={YES}]	
{NO }	
[,STATION={DCE}]	
{DTE}	
[,STATOPT={text string }]	V3R4 - V3R8
[,STATOPT={text string }]	(X.25 ODLC and NPSI starting
{(ALL,text string) }	with V3R9)
{(GROUP,text string,...LU,text string)}	
[,SUBADDR={YES}]	
{NO }	
[,SUBD=(n1,n2,n3,...nn)]	
[,SVCINN={number}]	
{0 }	
[,T1TIMER={time}]	
{0 }	
[,TAXUNIT={0 }]	(Default for non-fast connect GATE function)
{32 }	
{64 }	(Default for fast connect)
{128}	
[,T3=(YES,timer)]	(X.25 ODLC and NPSI V3R4 and
(NO)	later)
[,T4=(timer)]	
(60)	
[,TDTIMER=timer]	
[,TPTIMER=timer]	

Table 5-10 (Page 5 of 5). X.25 Keywords for X25.MCH Statement

Keyword	
[,TRAN={ODD }] {EVEN } {NO } {MARK } {SPACE} {USER1} {USER2}	
[,TRANSFR={count}]	(X.25 ODLC and NPSI V3R5 and later)
[,VCCINDX=index]	(X.25 ODLC and NPSI starting with V3R9)
[,VCID={YES}] {NO }	
[,VMODTAB=(v0,v1,v2,...vm)]	
[,VTMKYWRD=(targ_name,kywrd_name,kywrd_value)]	(X.25 ODLC and NPSI starting with V3R9)
[,XMONLNK={YES}] {NO }	
[,X21INACT={time}]	
[,X21NTWK={NO }] {OLD } {1984}	
[,X21RTYCT={count}]	
[,X21RTYT0={time}]	
[,X21SW={YES }] {NO } {T70DXE }	(V3R7 and later)

Table 5-11. VTAM Keywords for X25.MCH Statement

Keyword	
[,INNPOR=...]	(X.25 ODLC and NPSI V3R4 and later)

symbol

An optional label. When coded, it overrides the automatic resource name generation for the corresponding NCP line. The automatic name generation creates:

pLxxx

See “Resource Naming Conventions” on page 5-2 for information on default naming conventions.

ADDRESS={xxxx},

Specifies the duplex line interface address in 1–3 decimal digits. Valid values range from 0–63 for the 3720 and 0–895 for the 3745. The value specified must be in the range allowed by the NCP USGTIER value specified on the BUILD statement. For example, line addresses 32–63 are not allowed on a 3720 with a USGTIER of 1.

Note: If several resources are included in the NCP generation, such as Network Routing Facility (NRF), you might have to use three decimal digits for the address. For example, if the line address is 1, code ADDRESS=001. The X25.PREFIX keyword can also be used on the BUILD statement.

See the *NCP, SSP, and EP Resource Definition Reference* for the ranges of addresses corresponding to NCP USGTIER values.

V3R5 and Later:

For connections through an HPTSS, the ADDRESS keyword can specify the duplex line interface address in up to 4 decimal digits. Valid values range from 1024 to 1039 for 3745 models, and 1028 to 1031 for 3745-1xx models.

Note: The processing for building resource names at system generation has been modified for HPTSS processing.

X.25 ODLC and NPSI V3R8 and later with 3746 Model 900:

For connections through a CSS, the ADDRESS keyword can specify the duplex line addresses in 4 decimal digits. Valid values are in the range of LIC11 or LIC12 addresses.

Notes:

1. The address specified on the ADDRESS keyword is not the X.25 MCH address but the address of the outboard data link control (ODLC) physical line.
2. For X.21 switched connections, only LIC 12 addresses are allowed.

FRMLGTH=*length*,

Specifies the maximum frame length, in bytes, that flows over the physical circuit. *length* is the data packet maximum length added to the packet header length. Valid values range from 35–4100 (according to the network subscription). Packet header length is:

- 3 bytes for packet modulo 8
- 4 bytes for packet modulo 128

Note: The minimum value for an MCH that supports subarea node virtual circuits is 131.

For X.25 ODLC, the default for FRMLGTH is 131. There is no default for NPSI.

LCGDEF={{(*lcg*,*lcnhi*)
 {{(*lcg*,*lcnhi*),...}}

Specifies the greatest logical channel number in decimal (*lcnhi*) for each LCG. The *lcg* must be in decimal also.

You must define at least one pair of values; however, you can define up to 16 pairs in any order. Define only the LCGs that are actually used. LCNHI must appear in the LCN keyword of the X25.VC or X25.LINE statement that defines the last VC in the group.

The LCGDEF keyword is not required for X.25 ODLC and is ignored if coded.

MWINDOW=*window*

V3R6 and Earlier: Specifies the frame window size to be used by the X.25 Link Access Protocol (LAP) level 2. Valid values range from 1–7. This parameter is mandatory unless it has been specified in the BUILD statement. If it is specified in the X25.MCH statement and in the BUILD statement, the value specified in the X25.MCH statement overrides the value specified in the BUILD statement.

X.25 ODLC and NPSI V3R7 and Later:: If you coded MMODULO=128, the valid ranges for MWINDOW are from 1–127.

```
[,ACCOUNT={NO
           {YES
           {(LLC0,LLC2,LLC3I,LLC3B,LLC4,LLC5)}}
```

V3R4 and Later: Specifies whether all virtual circuits dedicated to the MCH are reporting NPM accounting data.

NO Specifies that reporting does not occur. The ACCOUNT keyword defaults to NO if the X25.BYTTHROD and X25.SEGTHROD keywords are not specified on the BUILD statement.

YES Specifies that all virtual circuits on this MCH report accounting data. You must support all LLCs on this MCH statement. The ACCOUNT keyword defaults to YES if X25.BYTTHROD and X25.SEGTHROD are specified on the BUILD statement.

(LLC0,LLC2,LLC3I,LLC3B,LLC4,LLC5)

Specifies the list of virtual circuit LLC types on this MCH that will report accounting data. This list of LLCs must be in accordance with the LLCs listed on the LLCLIST keyword of the X25.MCH statement.

Notes:

1. LLC3I stands for LLC type 3 INN (subarea node traffic), and LLC3B stands for LLC type 3 BNN (peripheral node traffic).
2. ACCOUNT=YES or ACCOUNT=(LLC...) is valid only if the X25.BYTTHROD and X25.SEGTHROD keywords are specified on the BUILD statement.

For an X.25 ODLC MCH, the only valid values are NO, YES, LLC3I, and LLC3B. For X.25 ODLC, LLC3I and LLC3B are the same as YES.

```
[,ACTIVTO=timer]
```

Specifies the value of the inactivity timeout in seconds. An SVCSC is cleared and the PU is made inoperable number of seconds specified by the ACTIVTO keyword lapse if either BRKCON=CONNECTO or if BRKCON=NOWNERTO is specified and SHM is not in effect. The connection must be reestablished using a VARY NET,DIAL command after such an inactivity timeout. Valid values range from 0–5400 seconds. This keyword applies to SVCSC only, and is valid only when SVCINN has a nonzero value.

The default is 0, which indicates that no timeout applies to this MCH.

For X.25 ODLC, the only valid value for ACTIVTO is 0.

```
[,ADDIFG={NO
          {YES}}
```

X.25 ODLC and NPSI V3R8 and Later:

Specifies whether the gap between frames should be made larger. This helps to keep the adjacent adapter from getting overrun errors because larger inter-frame gaps cause the frame rate to decrease.

Coding ADDIFG =YES is recommended when the adjacent adapter to which this adapter is transmitting is experiencing overrun errors.

```
[,ADJLINK={text string
           {ALL,text string}
           {GROUP,text string,...LU,text string}}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJLINK keyword and its text string are copied to the LINE statement generated by the X25.MCH statement.

If **ALL**, *text string* is coded, then the ADJLINK keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the ADJLINK keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJLINK on the PU and LU statements generated by the X25.MCH statement, code

```
ADJLINK=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJLINK=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJLINK=YYYY
```

will be generated following the LU statement.

```
[,ADJNODE={text string    }]  
  {ALL,text string}  
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJNODE keyword and its text string are copied to the PU statement generated by the X25.MCH statement.

If **ALL**, *text string* is coded, then the ADJNODE keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the ADJNODE keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJNODE on the PU and LU statements generated by the X25.MCH statement, code

```
ADJNODE=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJNODE=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJNODE=YYYY
```

will be generated following the LU statement.

```
[,AEFBYTE=(aa,bb,...)]
```

Specifies the last byte of the Called Address Extension Facility (AEF) (code: X'C9') contained in the Incoming Call packet used for the CTCP selection function. *aa,bb* represents the last byte, and must be coded in hexadecimal and in 2 digits. The valid value range is 00–FF.

Notes:

1. You must code the AEFBYTE and AEFCTCP keywords together.
2. FTPI=NO, CONNECT=NO and GATE=GENERAL are mandatory with the AEFBYTE and AEFCTCP keywords.

[,AEFCTCP=(*ii,jj,...*)] Specifies the CTCP number to be selected. *ii,jj* are the CTCP numbers and must be coded in decimal and in 2 digits. Up to 28 CTCPs can be selected. The AEFCTCP keyword is mandatory with the AEFBYTE keyword. Valid values are 00–27. You must code the AEFCTCP and AEFBYTE keywords together.

A correspondence is established between AEFBYTE and AEFCTCP. For example, if the last byte of the Called AEF in the Incoming Call packet is *bb*, then CTCP *jj* is selected.

Note: FTPI=NO, CONNECT=NO and GATE=GENERAL are mandatory with the AEFBYTE and AEFCTCP keywords.

[,AGGPRI={*text string* }]
 {ALL,*text string*}
 {GROUP,*text string*,...LU,*text string*}

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the AGGPRI keyword and its text string are copied to the LINE statement generated by the X25.MCH statement.

If **ALL**,*text string* is coded, then the AGGPRI keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the AGGPRI keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “*targ_name,value*” group. For example, to code AGGPRI on the PU and LU statements generated by the X25.MCH statement, code

```
AGGPRI=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* AGGPRI=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* AGGPRI=YYYY
```

will be generated following the LU statement.

[,ANS={CONTINUE|CONT}]
 {STOP }

Specifies the ANS keyword that is copied in the NCP PU statement for the defined physical circuit. CONTINUE can be abbreviated as CONT. Code ANS=CONTINUE or ANS=CONT if CONT is coded in any of the X25.VC, X25.FCG, or X25.PU statements associated with the dependent virtual circuits. Also, code ANS=CONT if XMONLNK=YES is specified. The default value is CONT.

```
[,BRKCON={CONNECTO}]
      {NOWNERTO}
      {NONE   }
```

Specifies that NPSI breaks a switched connection if a line is idle (not sending or receiving data) for a period of time specified by ACTIVTO on the GROUP statement. BRKCON is valid only if you code DIAL=YES and PUTYPE=4.

If you code BRKCON=CONNECTO or BRKCON=NOWNERTO, NPSI starts the inactivity timer after each data sent or received, and breaks the connection after ACTIVTO timeout. The default is BRKCON=NONE.

For X.25 ODLC, the only valid value for BRKCON is NONE.

```
[,CAUSE=DCE]
```

Determines the type of cause and diagnostic byte that NPSI sends to the network in Reset and Restart packets. This parameter overrides the selected CAUSE keyword on the X25.NET statement for this MCH.

When CAUSE=DCE is coded on the X25.MCH statement, you must code STATION=DCE and GATE=GENERAL, and only permanent virtual circuits using LLC type 4 can be used on this MCH.

Note: CAUSE=DCE cannot be coded for fast connect or FTPI.

```
[,CCXDELAY={timer}]
      {NONE }
      {2   }
```

The CCXDELAY keyword only applies to casual connection on SVC. CCXDELAY specifies the amount of time that NPSI delays its initial transmission after answering an incoming call on a QLLC peripheral link. CCXDELAY is valid only if you code LLC3 on the LLCLIST. Valid values range from 1–255. The default value is 2 seconds. If no casual connection is to be established on this MCH, code CCXDELAY=NONE.

The CCXDELAY keyword is ignored by X.25 ODLC.

```
[,CLRSESSQ={PU,key_value      }
            {LU,key_value      }
            {LU,key_value,PU,key_value}
```

This keyword and its *key_value* will be generated as a keyword on the generated PU statement or LU statement or both, to be processed by VTAM.

```
[,CMTKYWRD=(targ_name,kywrd_name,kywrd_value)      ]
```

Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrd_name is the name of the keyword to be generated.

kywrd_value is the value to be coded as an operand for the keyword.

```
[,CONNECT={NO  }
           {YES }
           {CUD0}
           {SUBD}
```

Specifies whether the fast connect function of NPSI is used on this physical circuit.

NO Specifies that fast connect is not used on this physical circuit. It is the default value.

YES Specifies that fast connect is used on this physical circuit. CONNECT=YES is used when only one CTCP is used on this fast connect MCH.

CUD0

Specifies that fast connect with multiple CTCPs is used on this physical circuit, and that the CTCP is selected through CUD0. When CONNECT=CUD0 is specified, the CUD0 and CTCP keywords must be coded on the X25.MCH statement. Then the following optional keywords can be specified on the X25.MCH statement:

- LUNAME
- MODETAB
- VMODTAB

Specify a value for each *unique* CTCP value. Thus, if 10 parameters were specified on the CTCP keyword, but there were only two unique values (00 and 01), these keywords should contain only two corresponding values.

The parameters are specified in a position corresponding to the CTCP number. The parameter in the first position corresponds with CTCP 0, the second with CTCP 01, and so on.

SUBD

Specifies that fast connect with multiple CTCPs is used on this physical circuit and the CTCP is selected through subaddressing. When CONNECT=SUBD is specified, the SUBD and CTCP keywords must be coded on the X25.MCH statement. Then the following optional keywords can be specified on the X25.MCH statement:

- LUNAME
- MODETAB
- VMODTAB

Specify a value for each *unique* CTCP value. Thus, if 10 parameters are specified on the CTCP keyword, but there are only two unique values (00 and 01), these keywords should only contain two corresponding values.

The parameters are specified in a position corresponding to the CTCP number. The parameter in the first position corresponds with CTCP 0, the second with CTCP 01, and so on.

V3R6 and Later: Code CONNECT=NO if INTWLINK=DIALDTE is coded on the X25.MCH statement for X.25/ISDN support.

Notes:

1. When subaddressing is used for CTCP selection, if an incoming call is received without a called DTE address (no subaddressing digit), the call is treated as if there was a subaddressing digit of 0.
2. If CONNECT=YES, CUD0, or SUBD is coded on the X25.MCH statement, the LOGAPPL keyword is optional.
3. PVCs are not allowed on fast connect MCHs.

[,CTCP=(m1,m2,m3,...,mn)]

The CTCP keyword is used mainly to select a CTCP, but it can also be used in conjunction with the CUD0 keyword to select an LLC or both an LLC and a

CTCP. Valid values to select a CTCP range from 0–27 (in decimal). Valid values to select an LLC are 80 for LLC type 0, 82 for PSH, 83 for QLLC, and 85 for PAD. You can code CTCP depending on the CUD0 keyword:

- Omit the CUD0 keyword. Do not code the CTCP keyword.
- Code CUD0=ALL. This provides a set of default values. Do not code the CTCP keyword.
- Code CUD0=(set of numbers). CTCP numbers or LLC numbers must correspond to the values coded in the CUD0 or SUBD keywords.

Values 00–27 select the CTCP. Values 80, 82, 83, and 85 select LLC types 0, 2, 3, and 5, respectively. (LLC type 4 is selected implicitly by coding a CTCP number.)

Note: The LLC selected by CTCP coding (corresponding to values 80, 82, 83, and 85) should also be specified in LLCLIST. Also, if CTCP selects one or several CTCPs (values 00–27), LLC4 should be specified in LLCLIST.

For example:

- If CTCP=(82) is coded, also code LLCLIST=(LLC2).
- If CTCP=(02) is coded, also code LLCLIST=(LLC4).
- If CTCP=(80,03,85) is coded, also code LLCLIST=(LLC0,LLC4,LLC5).

If the LLC deducted from CTCP coding is not coded in LLCLIST, NDF will set it anyway.

In the case of fast connect, you must code the CTCP keyword if the SUBD keyword is coded.

V3R8 and Later: If FTPI=PVC is coded, CUD0 and SUBD must not be coded. The CTCP keyword must still be coded with the authorized CTCPs.

```
[,CUD0={ (n1,n2,n3,...,nn,NULL) }
        (n1,n2,n3,...,nn,ALL) }
  {ALL }
```

Specifies the value of the first byte of the CUD that NPSI uses to select the following:

- A CTCP when CUD0=ALL is coded or when CUD0 is omitted (see the first and second items below).
- A CTCP and/or an LLC in conjunction with the CTCP keyword when CUD0=(n1,n2...) is coded (see the third and fourth item below).

CUD0 is allowed only if GATE=GENERAL is coded.

Note: To code CUD0 and the CONNECT keyword (for Fast Connect), see page 5-54. To code CUD0 and the FTPI keyword, see page 5-59.

You can code CUD0 and the CTCP keyword in the following four ways:

1. Omit the CUD0 keyword. Only one GATE CTCP (0) is selected. It is selected if there is no CUD (NULL CUD), or if CUD0 contains X'00' or X'C4', or any value from X'02'–X'2F'. Do not code the CTCP keyword.
2. Code CUD0=ALL. This provides a set of default values. Do not code the CTCP keyword.
 - The first GATE CTCP (0) is selected when CUD0 is equal to X'02' or X'C4'.

- The second GATE CTCP (1) is selected when there is no CUD (NULL CUD), or if CUD0 contains X'00', or any value from X'03'–X'2F'.
3. Code CUD0=(set of numbers). The combination of CUD0=(set of numbers) plus the CTCP keyword will select the GATE CTCP or LLC. Only the values coded will be used.
 4. Code CUD0=(set of numbers,ALL).

Starting with V3R9: This provides maximum flexibility for selecting the GATE CTCP or LLC type. Specified values override the default values shown in CUD0=ALL. Unspecified values take the default values shown with CUD0=ALL. The value ALL must be the last one specified.

Values range from X'00'–X'FF' plus NULL, with NULL designating an empty CUD field. If one of the values is to be rejected, the CTCP issues the CLEAR command at the reception of the call-in that is not valid.

Examples of CUD0 and CTCP coding are:

```
CUD0=(00,01,02,03,04,C1,CC,71,61,NULL)
CTCP=(00,00,01,00,01,80,80,85,85,01)
```

The result of this coding is:

- The first CTCP is selected when CUD0 is coded with X'00', X'01', and X'03'.
- The second CTCP is selected when CUD0 is coded with X'02', X'04', and NULL.
- LLC type 0 is selected by X'C1' and X'CC'.
- LLC type 5 is selected by X'71' and X'61'.

```
CUD0=(00,01,02,03,NULL,04,ALL)
CTCP=(00,00,01,01,02,02)
```

The result of this coding is:

- The first CTCP (0) is selected when CUD0 is coded with X'00', X'01', including the non-overridden default value of X'C4'.
- The second CTCP (1) is selected when CUD0 is coded with X'02', or X'03', including the non-overridden default values of X'05'–X'2F'.
- The third CTCP (2) is selected when CUD0 is a NULL CUD or coded with X'04'.

Specify or omit the leading zero in the CTCP and CUD0 keywords. This allows you to align the corresponding subkeywords by coding them one above the other.

[,DBIT={YES}]
{NO }

Specifies whether the D-bit is to be used. The D-bit is an indicator in the packet header of data packets flowing to and from non-SNA DTEs supported by type 0 virtual circuits. It confirms that a packet was received at its destination. The default value is NO. If you code DBIT=YES, the following events occur:

- The D-bit is set in the last packet of a packet sequence built from an outgoing PIU carrying any of the definite response bits.

- The definite response bit is set on in an incoming O/C PIU when the corresponding incoming packet (or the last packet of a packet sequence) has the D-bit on, and, if at BIND time, the definite response was requested on the secondary-to-primary flow.
- If MBITCHN=YES is also coded, the CPS series is processed as described in “NPSI Support of RU Chaining for Long Non-SNA Messages” on page 3-42.

```
[,DEFPRI={NO  }]  
          {LOW  }  
          {NORMAL}  
          {HIGH1}  
          {HIGH2 }
```

V3R4 and Later: Specifies the QOS priority value given to all the virtual circuits on the MCH. The default value is NO.

Note: You can code DEFPRI only if GATE=GENERAL is coded on the X25.MCH statement.

DEFPRI=NO means QOS processing is not done on this MCH. When DEFPRI is coded with a value other than NO, this value represents the transmission priority affected on the following:

- Non-GATE peripheral node leased traffic
- GATE leased traffic if DEFPRI is not coded on X25.VC and X25.LINE
- GATE switched traffic without an OSI priority facility in the Call packets
- GATE switched traffic with an OSI priority value that is reserved (X'0F' to X'FE') or unspecified (X'FF')

```
[,DIRECT={YES}]  
          {NO }
```

X.25 ODLC and NPSI V3R4 and Later: Specifies whether the physical circuit is directly attached to a communication controller or to a peripheral node. The default value NO specifies modem attachment.

Internal clocking is provided by coding DIRECT=YES. Although DIRECT=YES is coded, the NPSI generated statements still appear to have external clocking. However, because NPSI (not NCP) handles these lines, the appearance of this NCP generation statement is not important to internal clocking. When you code DIRECT=YES, NPSI can correctly instruct the hardware scanner to use internal clocking. This is the only generation change required to provide internal clocking.

```
[,DSABLTO=timer]
```

Specifies the value of the disable timeout to the nearest tenth of a second. This timer value is set by the modem and represents the maximum delay the DTE expects before the DSR signal ends after the DISABLE command has been issued. The range is 0.1 second–1632.0 seconds in increments of 0.1 second; the default is 3.0 seconds.

```
[,DTETODTE={YES}]  
          {NO }
```

X.25 ODLC:

Valid only when ODLC=YES is specified.

Specifies whether the DTE is directly attached to another DTE that supports the ISO8208 DTE-to-DTE protocol. The default is NO.

If DTETODTE=YES is coded, when the caller is acting as the DCE outgoing calls will use the lowest available SVC even if that SVC is defined as

CALL=IN. It is recommended that you define only CALL=INOUT SVCs to prevent confusion.

[,ENABLTO=*timer*]

Specifies the value of the enable timeout to the nearest tenth of a second. This timer value is set by the modem, and represents the maximum delay during which the DTE expects the DSR signal to begin after the ENABLE command has been issued. The range of values is 0.1 second–1632.0 seconds in increments of 0.1 second; the default is 3.0 seconds.

The timer is also used for modem retrain and DSR recovery. You should specify a large enough value for this keyword so that the time during which Data Set Ready (DSR) or Clear To Send (CTS) are down is exceeded. The DSR or the CTS might be down temporarily for tests or malfunctions.

This procedure keeps the physical circuit and all its virtual circuits up despite temporary CTS or DSR down conditions induced by the network.

[,FTPI={**NO** }]

{**YES** }

{**CUD0**}

{**SUBD**}

{**PVC** }

V3R4 and Later: Specifies whether FTPI is supported on the MCH. FTPI support is valid if GATE=GENERAL, SVCINN=0, SUBADDR=NO, CONNECT=NO, LLCLIST=LLC4, X21SW=NO, and VCID=YES are coded on the X25.MCH statement.

NO Specifies that FTPI is not used on this physical circuit. The default is NO.

YES Specifies that FTPI is used on this physical circuit. Only one CTCP (number 0) is used on this MCH. Therefore, you must code an X25.CPL statement with CTCPNO=0.

CUD0 Specifies that FTPI with multiple CTCPs is used on this physical circuit, and that the CTCP is selected through CUD0. Many CTCPs can be defined. The CTCP and CUD0 keywords must be coded. The CUD0 keyword is paired with the CTCP keyword to create a correspondence table between two positional operands. These keywords must have the same number of subvalues. Each CTCP subvalue refers to a previous X25.CPL definition.

SUBD Specifies that FTPI with multiple CTCPs is used on this physical circuit, and the CTCP is selected through subaddressing. The CTCP and SUBD keywords must be coded. The SUBD keyword is paired with the CTCP keyword to create a correspondence table between two positional operands. These keywords must have the same number of subvalues. Each CTCP subvalue refers to a previous X25.CPL definition.

V3R6 and Later: FTPI=NO must be coded if INTWLINK=DIALDTE is coded on the X25.MCH statement for X.25/ISDN support.

V3R8 and Later: FTPI=PVC can be coded to define an FTPI MCH with PVCs only. The CTCP keyword on the X25.MCH statement must also be coded.

NPSI V3R7 and NCP V7R2 and Later: NPSI APAR IR27307 must be applied for FTPI to work.

Note: The VTAM keywords, which are usually allowed on the X25.MCH statement (LOGAPPL, MODETAB...), are not authorized on FTPI MCHs. The VTAM keywords coded on the X25.CPL statement are taken into account for the corresponding CTCP number.

[,FTPITH={0|1024}]

Starting with V3R9: Specifies the threshold for queued outbound packets per VC. Valid values are 0 to 1024. The default value for the FTPITH keyword is 0, which disables this function. Once this threshold is exceeded, a timer of length FTPITHTO is started and if the outbound packet count for the VC does not drop below this threshold before the timer elapses, the VC will be reset.

[,FTPITHTO={0|255}]

Starting with V3R9: Specifies the timeout period (in seconds) in which the outbound packet count can exceed the threshold (FTPITH) before the VC is reset. Valid values are 0 to 255. The default value for the FTPITHTO keyword is 0, which disables this function.

[,GATE={DEDICAT}
{GENERAL}
{NO }]

Specifies whether the GATE or the DATE function is supported on this physical circuit. Code:

- GATE=DEDICAT for the DATE function.
- GATE=GENERAL for the GATE function (with or without fast connect or FTPI).
- GATE=NO if neither the GATE nor the DATE function is used.

If you code GATE=GENERAL, you can also code the SUBADDR, CTCP, CUDO, and SUBD keywords. The default value is NO.

Note: You must code GATE=GENERAL to initiate FTPI processing.

V3R6 and Later:

1. Do not code GATE=DEDICAT if INTWLINK=DIALDTE or INTWLINK=DIALPSDN is coded on the X25.MCH statement for X.25/ISDN support.
2. Do not code GATE=GENERAL if INTWLINK=DIALDTE is coded on the X25.MCH statement for X.25/ISDN support.

For X.25 ODLC, the only valid value for the GATE keyword is NO.

[,GATEINFO={NO }
{YES}]

Starting with V3R9: Specifies whether an SNA negative response is returned to commands from the CTCP that are rejected. Specifying YES suppresses the negative response. The default is NO.

[,IDBLKC={ccc}]

Specifies the value of IDBLK for PCNE connections. Valid values for *ccc* range from X'062' to X'06E'. This keyword allows improved concurrent use of GATE, PAD, and PCNE connections on the same MCH.

[,IDBLKG={ggg}]

Specifies the value of IDBLK for GATE connections. Valid values for *ggg* range from X'062' to X'06E'. This keyword allows improved concurrent use of GATE, PAD, and PCNE connections on the same MCH.

[,IDBLKLCG={(*lcg*,*idblock*)
 {((*lcg*,*idblock*),(*lcg*,*idblock*),...))}]

V3R8 and Later: Specifies the relationships between X.25 LCGs and SNA IDBLKs for connections with DTEs that do not provide their identifications (PCNE, PAD, GATE).

The value for *lcg* is the X.25 LCG number. Valid values for *lcg* are 0–15 (in decimal). Any *lcg* values coded on the IDBLKLCG keyword must also be coded on the LCGDEF keyword.

idblock is the SNA IDBLK to be associated with the X.25 LCG. Valid values for *idblock* are 003 and 062–06E (in hexadecimal).

Note: When you code the IDBLKLCG keyword, you cannot code the IDBLKC, IDBLKG, and IDBLKP keywords on this X25.MCH statement.

[,IDBLKP={*ppp*}]

Specifies the value of IDBLK for PAD connections. Valid values for *ppp* range from X'062' to X'06E'. This keyword allows improved concurrent use of GATE, PAD, and PCNE connections on the same MCH.

[,INTFAC={NO }]
 {YES}

Specifies whether NPSI interprets and uses the flow control parameters of the Call Connected packet under GATE or DATE. The default is NO.

[,INTWLINK={NO }]
 {DIALDTE }
 {DIALPSDN}

V3R6 and Later: Specifies whether the MCH is dedicated to switch access to a PSPDN. If INTWLINK=NO, the MCH does not support interworking connections. If INTWLINK=DIALDTE, the MCH is dedicated to outgoing connections from NPSI to PSPDN through ISDN and 7820 (dial-in-by-the-DTE). If INTWLINK=DIALPSDN, the MCH is dedicated to incoming connections from PSPDN to NPSI through ISDN and 7820 (dial-out-by-the-PSPDN). The default is NO.

For DIALDTE or DIALPSDN:

- You cannot code the PHYSRSC keyword on the X25.MCH statement. This enables the MCH and associated virtual circuits to process as independent resources rather than as a complex link.
- You must code SPEED=64000 on the X25.MCH statement to set the data rate for the physical circuit.
- The DATE function is not allowed on the MCH, so you cannot code GATE=DEDICAT on the X25.MCH statement.
- X21SW=NO is required, because the physical line cannot be an X.21 switched line.
- PVCs are not allowed.

For DIALDTE, STATION=DTE is mandatory on the X25.MCH statement. Only outgoing SVCs are allowed on the MCH. Code SHM=NO on the MCH to indicate that SHM is not used on the SVCSC lines. The following restriction applies:

- The GATE function is not allowed on the MCH, so you cannot code GATE=GENERAL on the X25.MCH statement. Consequently, LLC4 cannot be coded in LLCLIST. Use either the default or NO on CONNECT and FTPI keywords.

For DIALPSDN, the following restrictions apply:

- Only one incoming SVC is allowed on the MCH.
- The definition of a SVCSC line is not allowed on the MCH.

For X.25 ODLC, the only valid value for the INTWLINK keyword is NO.

[, ISTATUS={ACTIVE }]
{INACTIVE}

Specifies whether the line, physical units, and logical units are activated following a VTAM first start or cold restart. There is no default value for this keyword. This keyword and parameter are reported on the generated NCP LINE statements.

For DATE or GATE, ISTATUS=ACTIVE is forced by NPSI generation on the LUs generated for this MCH. In any other case, ISTATUS=INACTIVE is forced on the LU generated by this X25.MCH statement.

[, ITRACE={YES}]
{NO }

Specifies whether LAP internal trace is performed on this physical circuit. The default is YES.

For X.25 ODLC, the default is NO and is the only valid value for the ITRACE keyword.

[, LCN0={NOTUSED}]
{USED }

Specifies whether logical channel number 0 of the logical channel group 0 designates a virtual circuit. The default value is USED.

Note: If you specify USED, LCN=0 must be defined in a subsequent X25.LINE or X25.VC statement.

For X.25 ODLC, the default is NOTUSED and is the only valid value for the LCN0 keyword.

[, LLCLIST=(LLC0, LLC2, LLC3, LLC4, LLC5)]

Specifies the types of SVCs that are supported on this physical circuit. LLCLIST is mandatory if you want to associate any SVCs with this physical circuit; however, it is not required if only PVCs are associated with this physical circuit. These keyword values are not positional. Code one value for each SVC type as follows:

- Code LLC0 if one or more SVCs can connect to a non-SNA DTE, and these circuits do not require GATE or PAD support (type 0 virtual circuits).
- Code LLC2 if one or more SVCs can connect to an SNA peripheral node with PSH protocol (type 2 virtual circuits).
- Code LLC3 if one or more SVCs can connect to an SNA peripheral or subarea node with protocol QLLC (type 3 peripheral or subarea node virtual circuit).
- Code LLC4 if one or more SVCs can require GATE support (type 4 virtual circuits).

V3R6 and Later: LLC4 is not allowed if INTWLINK=DIALDTE is coded on the X25.MCH statement for X.25/ISDN support.

- Code LLC5 if one or more SVCs can require transparent or integrated PAD support (type 5 virtual circuits).

Note: LLCLIST is not required for fast connect or FTPI MCHs. However, if LLCLIST is used, it must be coded as LLCLIST=LLC4.

For X.25 ODLC, the default is LLC3 and is the only valid value for the LLCLIST keyword.

```
[,{LLC0=(m1,m2...)},...]  
{LLC2=(m1,m2...)}  
{LLC3=(m1,m2...)}  
{LLC4=(m1,m2...)}  
{LLC5=(m1,m2...)}  
{LLCI=(m1,m2...)}
```

LLC0=(m1,m2,...)

Specifies the values in the subaddress field of Incoming Call packets that indicate that the calling DTE is a non-SNA DTE not requiring GATE or PAD support. *m* is any decimal digit from 0–9. This keyword is valid only when you code SUBADDR=YES.

LLC2=(m1,m2,...)

Specifies the values in the subaddress field of Incoming Call packets that indicate that the calling DTE is an SNA peripheral node attached with PSH protocol. *m* is any decimal digit from 0–9. This keyword is valid only when you code SUBADDR=YES.

LLC3=(m1,m2,...)

Specifies the values in the subaddress field of the Incoming Call packets that indicate that the calling DTE is an SNA peripheral node attached with the peripheral node QLLC protocol. *m* is any decimal digit from 0–9. This keyword is valid only when you code SUBADDR=YES.

LLC4=(m1,m2,...)

Specifies the values in the subaddress field of Incoming Call packets that indicate that the calling DTE requires GATE support. *m* is any decimal digit from 0–9.

LLC5=(m1,m2...)

Specifies the values in the subaddress field of Incoming Call packets that indicate that the calling DTE requires integrated or transparent PAD support. *m* is any decimal digit from 0–9. This keyword is valid only when you code SUBADDR=YES.

LLCI=(m1,m2...)

Specifies the values in the subaddress field of the Incoming Call packets, which indicate that the calling DTE requires switched subarea node. *m* is any decimal digit from 0–9. This keyword is valid only when you code SUBADDR=YES on the X25.MCH statement. If this feature is used, the caller must insert at least an extra digit at the end of the called DTE address in the DIALNO (NN field in DIALNO) of the PATH statement in the remote SMN.

Note: You must code different values for *m* in each of the LLC0, LLC2, LLC3, LLC4, LLC5, and LLCI keywords. This allows you to select a given LLC based on the subaddressing digit values.

```
[,{LOGAPPL=(appl0,appl1,...,applm)}
```

Specifies the application names corresponding to the CTCP numbers specified in the CTCP keyword. This causes VTAM to automatically generate a logon request to the specified application on behalf of the associated physical circuit LU when the LU is activated by VTAM. There is one unique subvalue for

X25.MCH

each of the subvalues of the CTCP keyword. NPSI provides no default for this keyword.

LOGAPPL is not valid if FTPI=CUD0 or FTPI=SUBD is coded on the X25.MCH statement. LOGAPPL is optional if CONNECT=YES, CONNECT=CUD0, or CONNECT=SUBD is coded on the X25.MCH statement.

[,LSPRI={YES}]
{NO }

Specifies whether link SESSION priority is given to peripheral lines (boundary virtual circuits working under LLC type 2 or LLC type 3) on this MCH. LSPRI=YES allows outbound PIU flow prioritization between the LUs from a peripheral node. LSPRI=NO is the default option.

[,LUNAME=(*luname0,luname1,...lunamem*)]

Specifies the names of the logical units associated with this MCH. There is one unique subvalue for each of the subvalues of the CTCP keyword. See "Resource Naming Conventions" on page 5-2 for information on default naming conventions.

[,MACB=(*labelx,labelr*)]

Specifies the labels of control blocks (UACBs). Control block labels are usually created by NPSI during generation. Code MACB to prevent label duplication with other resource names in the same generation. If you do not code this keyword, NPSI creates the UACB labels in the following formats:

X25AxxxX
X25AxxxR

where:

xxx is the physical circuit line address as coded in the X25.MCH ADDRESS keyword (1–3 digits).

[,MBITCHN={YES}]
{NO }

Specifies whether RU chaining is supported on outbound and inbound flow for non-SNA connections. When MBITCHN=YES is specified, RU chaining is supported. The default value is NO. MBITCHN=YES is not valid when GATE=DEDICATE is coded. For connections using GATE SVCs or Transparent PAD SVCs, MBITCHN=YES is valid, but the inbound flow is processed differently. See *X.25 NPSI Version 3 Host Programming* for information.

[,MBNOFULL={YES}]
{ACC}
{NO }

Specifies whether a data packet that is not full, with the M-bit on, is accepted. When MBNOFULL=NO is coded, a non-full data packet with the M-bit on is released, and the virtual circuit over which this packet was received is reset or cleared.

When MBNOFULL=YES is coded, a non-full data packet with the M-bit on is considered the last packet of a CPS. (The M-bit is ignored, and the inbound accumulation stops.)

When MBNOFULL=ACC is coded, a non-full data packet with the M-bit on is processed as a full data packet with the M-bit on (inbound accumulation continues).

The default value is NO.

Note: MBNOFULL is valid only if MBITCHN=NO and STATION=DTE are coded.

[,MMODULO={8|128}]

X.25 ODLC and NPSI V3R7 and Later:

Specifies the modulo value at frame level used by link access procedure balance (LAPB). You can specify MMODULO as 8 or 128. The default value for the MMODULO keyword is 8.

NPSI V3R8 and later: For X.25 lines connected through the CSS, the following rules apply:

1. A CLP can support up to four X.25 modulo 128 lines per pair of LIC slots, compared to 60 lines of another type.
2. Each X.25 modulo 128 line reduces the total connectivity by 15 for the other types of lines supported sharing the same CLP. For example, if one X.25 modulo 128 line is defined on a CLP (with two LICs), up to 45 non-X.25 modulo 128 lines such as SDLC, frame relay, or X.25 modulo 8 can also be defined.

[,MODETAB=(table0,table1,...tablem)]

Specifies the mode table (MODETAB) for the physical circuit LUs defined in this X25.MCH statement. There is one unique subvalue for each of the sub-values of the CTCP keyword. NPSI provides no default for this keyword.

Note: MODETAB is not valid if you code FTPI=CUD0, FTPI=YES, or FTPI=SUBD on the X25.MCH statement.

[,NCPGRP={label}]
{NEW }

Specifies the label of the NCP GROUP statement that includes the physical circuit in the NCP. If you do not specify this value, the name is created by X.25 generation. The default for this keyword depends on the network.

If you omit this keyword, the built-in label is:

pNETnm

If you specify NEW, a new NCP GROUP statement is created and named using the automatic naming convention. If you specify *label*, a new NCP GROUP statement is created that has the specified name.

Note: One NCP GROUP statement is automatically generated for the first MCH and another NCP GROUP statement is generated for subsequent MCHs. The first MCH is in a group by itself; the second and subsequent MCHs will be part of the second GROUP unless this keyword is coded.

If you want to have all the MCHs of a given generation in the same NCP GROUP, you must code the same name for NCPGRP on all X25.MCH statements.

See “Resource Naming Conventions” on page 5-2 for information on default naming conventions.

[,NDRETRY={count}]
{1 }

Specifies the number of times that the Np/Tp sequence is executed. Valid values range from 1–255; the default is 1.

For X.25 ODLC, the only valid value for the NDRETRY keyword is 1. See the description of the TPTIMER keyword for more information.

[,NETID={name,ANY}]

Specifies the association of a switched subarea support group (DIAL=YES and PUTYPE=4) to a network. This keyword is copied in the GROUP statements generated for SVCSC lines. It is used with SNI.

name corresponds to either NETID on the BUILD definition statement (native network), or NETID on the NETWORK definition statement (nonnative network).

X.25 ODLC and NPSI V3R4 and Later: The dynamic NETID keyword function has been extended to INN leased lines. Like SVCSC, the NETID keyword is sifted in the GROUP statements generated for INN PVCs.

If you code NETID=(*name*,ANY), the names specified for NETID on the X25.PU statements (PU type 4) related to this MCH may be different from the name specified for NETID on the X25.MCH statement.

```
[,NGFINC={text string      }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFINC keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.MCH statement.

If **ALL**,*text string* is coded, then the NGFINC keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the NGFINC keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFINC on the PU and LU statements generated by the X25.MCH statement, code

```
NGFINC=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFINC=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFINC=YYYY
```

will be generated following the LU statement.

```
[,NGFOPT={text string      }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFOPT keyword and its text string are copied to the PU statement generated by the X25.MCH statement.

If **ALL**,*text string* is coded, then the NGFOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the NGFOPT keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFOPT on the PU and LU statements generated by the X25.MCH statement, code

```
NGFOPT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFOPT=YYYY
```

will be generated following the LU statement.

```
[,NGFTXT={text string  }]
  {ALL,text string}
  {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFTXT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.MCH statement.

If **ALL**,*text string* is coded, then the NGFTXT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the NGFTXT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFTXT on the PU and LU statements generated by the X25.MCH statement, code

```
NGFTXT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFTXT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFTXT=YYYY
```

will be generated following the LU statement.

```
[,NPACOLL={MCHLINE,MCHPU,VCPU}]
  {NO,NO,NO  }
```

X.25 ODLC and NPSI V3R4 and Later: Specifies whether NPM performance data is reported. X.25 ODLC and NPSI support NPM and allow you the option of coding NPACOLL on the X25.MCH statement to interface with the NPM performance function. The default value is NO. NPM performance data is reported on the following resources:

- LINE MCH when you code MCHLINE (link level)
- PU MCH when you code MCHPU (packet level)
- PU virtual circuit when you code VCPU (specific DTE)

Notes:

1. You can code MCHLINE, MCHPU, and VCPU on NPACOLL only if you code NPA=YES on the BUILD statement. These parameters are optional.
2. If you specify VCPU, you must also code the NPPVCN keyword on the X25.MCH statement. For X.25 ODLC, NPPVCN is not required and is ignored if coded.
3. When this MCH is a bridge with XI, you cannot code MCHLINE.

[,NPADTEAD={number}]

X.25 ODLC and NPSI V3R4 and Later: Identifies the DTE address of the MCH. NPADTEAD is required when ACCOUNT=YES or ACCOUNT=LLCx is coded. No default is provided. The valid value is a decimal number of 1 to 16 digits.

Note: NPADTEAD must not be coded if accounting is not supported on this MCH (ACCOUNT=NO coded on the X25.MCH statement, or X25.BYTTHROD and X25.SEGTHROD not coded on the BUILD statement).

[,NPPVCN={number}]

V3R4 and Later: Specifies the number of virtual resource PUs, for both switched and leased links, on which you need NPM performance data reporting at the same time. Valid values range from 1 to 4096. The coded value cannot be greater than the number of virtual resource PUs defined on this X25.MCH statement. NPPVCN is mandatory if NPACOLL=VCPU is coded on this X25.MCH statement.

For X.25 ODLC, the NPPVCN keyword is ignored.

[,NPRETRY={count}]
 {Z }

Specifies the total number of transmissions of an I or U frame in a Tp timeout recovery. (NPRETRY corresponds to the CCITT N2 retry count.) Valid values range from 3–255; the default is 7. See the description of the TPTIMER keyword for more information.

[,ODLC={YES}]
 {NO }

X.25 ODLC: Code ODLC=YES to indicate that the MCH is controlled by NCP X.25 ODLC. The default is NO.

The ODLC keyword is valid only if LLC=LLC3 and only on IBM 3746 Model 900 CSS lines.

[,OWNER={owner_name }]
 {owner_name1,owner_name2,...}
 {owner_name1,PASS }
 {owner_name1,PASSTHRU }]

X.25 ODLC and NPSI V3R4 and later: Specifies the name of the owner for the physical MCH line. One or more owners can be defined and will be passed to VTAM on the LINE statement that gets generated. The value will not be sifted down to the X25.LINE or X25.VC statements under the X25.MCH statement.

If you want to have an owner name sifted down to the X25.LINE or X25.VC statement, then code PASS or PASSTHRU as the second value. In this case you can specify only one owner name. PASS or PASSTHRU will not be passed to VTAM.

```
[,PAD={INTEG }]  
    {TRANSP}  
    {NO }
```

Specifies whether the terminals supported by PAD can be attached through virtual circuits associated with this physical circuit. If PAD support is provided, this keyword also specifies the type of support as follows:

INTEG Specifies that NPSI provides integrated PAD support. This value can be specified even if GATE=DEDICAT is specified. In this case, SHUTD=NOINVCLR is mandatory.

TRANSP Specifies that NPSI provides transparent PAD support.

NO Specifies that NPSI does not provide PAD support. If NO is coded, LLC5 cannot be coded in the LLCLIST keyword, the LLC5 keyword cannot be coded in the X25.MCH statement, and LLC=LLC5 keyword cannot be coded in an X25.VC or X25.LINE statement for a PVC associated with this physical circuit. The default value is NO.

For X.25 ODLC, the only valid value for the PAD keyword is NO.

```
[,PADBRKCD={{cc1dd1,cc2dd2,...}}]  
    {0084,0085 }
```

Specifies the cause and diagnostic codes returned by the PAD when the BREAK key is entered and when PAD parameters 0702 or 0708 are specified. The valid range is specified by any 4 hexadecimal digits for both subvalues. Each subvalue is optional within the sublist. PADBRKCD is valid only if PAD=INTEG is coded on the X25.MCH statement. The default value ID is 0084,0085.

Note: When the network does not support the diagnostic code in the Reset packet, you should not use options 2 and 8 of PAD parameter 7.

```
[,PADCHECK={NO }]  
    {YES}
```

Specifies whether the PAD response to a *set PAD parameters* command must be analyzed by NPSI for a successful PAD parameter setting. If an *error PAD command* is returned by the PAD when the *set PAD parameters* command cannot be satisfied, the following occurs:

- If PADCHECK=YES, NPSI clears the SVC or resets the PVC and INOPs the station.
- If PADCHECK=NO, NPSI ignores the *error PAD command*.

PADCHECK can be coded only if PAD=INTEG is coded. The default value is YES.

```
[,PADINDX={index}]  
    {0 }
```

Specifies the index to the PADPARM keyword, which is specified on the X25.PAD statement. You must specify *index* on the X.25 PAD statement for this statement to be valid. Valid values for PADINDX are 1–99. The default is 0. PADINDX is valid only if PAD=INTEG.

Note: For migration purposes, do not code the default value 0, which refers to the default PAD parameter-string. The PADINDX keyword should not be coded for PADS that do not conform to CCITT Recommendation X.3.

```
[,PHYSRSC={YES}]
  {NO }
```

X.25 ODLC and NPSI V3R5 and Later: Code PHYSRSC=YES so that NCP and VTAM know of the association between this MCH and its associated virtual circuits. This ensures that NCP and VTAM process the MCH and its associated virtual circuits as one complex link (delaying virtual circuit link activation until the MCH link and the PU are active, rejecting MCH PU deactivation if the virtual circuits are still active, and displaying the MCH name when a virtual circuit is displayed). If PHYSRSC is not coded, there is no default value, and the MCH and associated virtual circuits are still processed as independent resources, not as one complex link.

Note: You cannot code PHYSRSC for an MCH supporting DATE, Fast connect, FTPI, or X.25/ISDN.

For X.25 ODLC, the default and only valid value for the PHYSRSC keyword is YES.

```
[,PKTMODL={128}]
  {8 }
```

Specifies the modulo used by packet protocol. Valid values are 8 or 128; the default value is 8. You must specify a modulo greater than all the values specified in the VWINDOW keywords of the X25.VCCPT statements related to this MCH.

Note: The PKTMODL and FRMLGTH keywords are related. For example, if you code FRMLGTH=35, PKTMODL can be either defaulted or coded as PKTMODL=8. If you code FRMLGTH=4100, you must code PKTMODL as PKTMODL=128 (default 8 is not acceptable).

```
[,PLPIGGYB={YES}]
  {NO }
```

Specifies whether piggybacking is used. If PLPIGGYB=NO is specified, each data packet is normally acknowledged with an RR packet. If PLPIGGYB=YES is specified, data packets are acknowledged with an RR packet when (PACKET WINDOW minus 1) data packets have been received.

The default for PLPIGGYB is YES.

```
[,PUNAME=puname]
```

Specifies the name of the physical unit associated with this MCH.

```
[,PWPROT={YES  }]
  {YESWOCC}
  {NO   }
```

Specifies whether password protection is performed on the virtual circuits related to this MCH using integrated PAD. YES and YESWOCC are only valid if PAD=INTEG is defined on this physical circuit. The default is NO. For more details, see the PAD description on page 4-40 of this document.

```
[,RESETPVC={NO  }]
  {YES }
```

X.25 ODLC and NPSI V3R7 and Later: Specifies whether to reset all active PVCs when the MCH is reinitialized. You can code RESETPVC=YES or RESETPVC=NO. The default is RESETPVC=NO.

If you code RESETPVC=YES:

- A PVC is reset (diagnostic X'A1') when the associated SNA link is activated and the MCH is already active.
- PVCs that are active at MCH initialization or reinitialization are reset with diagnostic X'A1'.

RESETPVC=YES is not authorized if GATE=DEDICATED is coded on the same X25.MCH.

If you code RESETPVC=NO, PVCs are not reset at PVC link activation or when MCH is activated or reinitialized.

For X.25 ODLC, the only valid value for the RESETPVC keyword is YES.

[,RESETUNB={YES}]
{NO }

Starting with V3R9: Specifies whether NPSI should send a RESET packet when a BIND or UNBIND is received from the application, or a data packet is received but the PCNE LU session is not bound. NO is the default.

When RESETUNB=YES is coded, MAXOUT=6 must be coded on the X25.VC or X25.PU statement that defines the PCNE PVC.

RESETUNB=YES is allowed only on the X25.LINE or X25.VC for LLC0 PVCs. RESETUNB=YES is not allowed when GATE=DEDICAT is specified on the X25.MCH statement.

[,RNRPKT={NO }]
{YES}

Specifies whether an RNR packet should be sent when data is received while in buffer shortage or when the destination route is blocked. If RNRPKT=YES is coded, NPSI sends an RNR packet when receiving inbound data packets when in an unsafe or danger status, even if the PLP window is not closed, or when the virtual route over which the packet must be sent is blocked. The default value is NO.

For X.25 ODLC, the only valid value for the RNRPKT keyword is NO.

[,RNRTIMER={0 }]
{nnn}

X.25 ODLC and NPSI V3R7 and Later: Specifies the delay in seconds for a received RNR packet to be compensated with an RR. You can specify a value from 0 to 200. The default value for *n* is 0.

If you code RNRTIMER with a nonzero value for *n*, a virtual circuit over which an RNR packet was received is reset (diag X'94') if no RR is received to compensate the RNR within *n* seconds.

If you code RNRTIMER=0 and no RR is received, the virtual circuit over which the RNR packet was received can remain in a holding state indefinitely.

An RNRTIMER keyword coded with a nonzero *n* is not authorized if GATE=DEDICATED is coded on X25.MCH.

For X.25 ODLC, the only valid value for the RNRTIMER keyword is 0.

[,ROLE={PRI}]
{NEG}

Specifies the data link control role that the station assumes on an SVC at contact time, when the remote partner is an SNA type 2.1 node. ROLE is valid only if you code LLC3 on the LLCLIST keyword. You can code ROLE as primary (PRI), which means that the station controls the session, or as negotiable (NEG), which means the station assumes either the primary or secondary role. The default value is NEG.

[,RPADPARM={YES}]
{NO }

V3R5 and Later: Specifies whether you want to have the READ PAD PARAMETER command sent to a DTE at call setup time for the integrated PAD function.

When you code RPADPARM=YES on the X25.MCH statement, NPSI initiates a DTE connection and sends the READ PAD PARAMETER command to the PAD.

X25.MCH

When you code RPADPARAM=NO on the X25.MCH statement, NPSI initiates a DTE connection, and does not send the READ PAD PARAMETER command to the PAD.

Note: RPADPARAM can be coded only if PAD=INTEG is coded on the X25.MCH statement.

The default value is YES if PAD=INTEG is coded.

[,SDRTCNT=*count*]

Specifies the number of retries made when the SHM reconnection cannot be established. The valid range is from 1–255; the default value is 0 (no retries).

[,SDRTIME=*time*]

Specifies the number of seconds between two retries of SHM reconnection. The valid range is from 1–255; the default value is 10 seconds.

[,SHM={YES }
{NO }]

Specifies whether SHM is used on the SVCSC lines dependent on this MCH. When SHM=NO is coded, no lines use SHM. When SHM=YES is coded, SVCSC connections on this MCH may use SHM. The default value is YES.

V3R6 and Later: SHM=NO is required if INTWLINK=DIALDTE is coded on the X25.MCH statement for X.25/ISDN support.

For X.25 ODLC, the default and only valid value for the SHM keyword is NO.

[,SHUTD={INVCLR }]
{NOINVCLR}

Specifies whether a shutdown request, sent to the LU that represents an integrated PAD virtual circuit, causes any action (INVCLR) or not (NOINVCLR). This action, for SVCs, is the sending of an Invitation to Clear PAD message, or for PVCs, the sending of a Reset packet. This keyword is valid only when PAD=INTEG is coded. The default is INVCLR (Invitation to Clear).

[,SPAN=*name*]

Specifies that the NetView program operator's access to the virtual circuit line is restricted. This specification is copied to the LINE, PU, and LU statements generated by the X25.MCH statement.

[,SPEED={*rate* }]
{9600 }
{245760 }
{256000 }

Specifies the data rate for the physical circuit in bits per second (bps). The default value is 9600. The basic range is from 1200–256 000 bps.

When DIRECT=YES is coded on the X25.MCH statement, only the following values are valid:

- For an IBM 3720: 1200, 2400, 4800, 9600, 19200
- For an IBM 3745 (all models):
1200, 2400, 4800, 9600, 19200, 38400, 55855, 245760

For LIC 4 dynamic configuration with all models of the 3745, and X.21 MCH lines with modem clocking (DIRECT=NO), the keyword value is used to select a pattern corresponding to one of the internal clocking values allowed in the controller. This pattern is forwarded to the scanner at MCH line activation and allows dynamic configuration of the LIC 4.

Table 5-12 on page 5-73 shows the internal clock speed and LIC4 configuration for the span of available line speeds.

Table 5-12. Internal Clock Speed and LIC4 Configuration for Span of Available Line Speeds

SPEED Keyword Value	Internal Clocking Selected	LIC4 Configuration Selected
$1200 \leq v < 19200$	9600 bps	four lines
$19200 \leq v < 144000$	55 855 bps	one line wideband
$144000 \leq v \leq 256000$	245 760 bps	one line wideband + high speed

See the *NCP, SSP, and EP Resource Definition Reference* for more information about the use of SPEED.

Note: The SPEED and T1TIMER keywords act on the piggybacking mechanism at the frame level, delaying the frame acknowledgment at the link level (X.25 T2 timer delay).

V3R5 and Later:

For HPTSS support, when you code DIRECT=YES on the X25.MCH statement, only the following values are valid for an IBM 3745 (all models):

- 245760
- 1474560
- 1843200

In this case, the default value is 245760.

For HPTSS support, when you code DIRECT=NO on the X25.MCH statement, you can code any speed value from 1200 to 2048000 on the SPEED keyword.

In this case, the default value is 256000.

V3R6 and Later:

For ISDN support, when you code INTWLINK=DIALDTE or DIALPSDN on the X25.MCH statement, you must code SPEED=64000.

X.25 ODLIC and NPSI V3R8 and Later:

Specifying SPEED is mandatory for X.25 with the IBM 3746 Model 900. For X.21 leased lines, the speed can be up to 2048 Kbps (for LIC12). For X.21 switched access to ISDN through the 3746 Model 900, you must code SPEED=64000, 384000 or 1500000.

If you code SPEED=64000 or SPEED=384000 for X.21 switched access to ISDN with the Hitachi TA, that value is used during the calling phase and during the data transfer phase. If you code SPEED=1500000, the speed during the calling phase can be 384 Kbps and the speed during the data transfer phase can be 1500 Kbps. Therefore, for X.21 access to ISDN with the Hitachi TA, code SPEED=64000 or SPEED =384000 only if both calling phase and data transfer phase use the 384 Kbps rate.

For more information see the *NCP, SSP, and EP Resource Definition Reference*.

[,SPNQLLC={YES}]
 {NO }

Specifies whether this MCH is connected to a network that supports an SDLC PAD that does not provide the standard QLLC interface. The effect of this keyword is to set X'42000000' in the CUD field rather than X'C3' for QLLC outgoing calls. SPNQLLC also accepts X'62' as the first byte of the CUD for incoming calls rather than X'C3'. The address portion of the Q commands will be set to X'EF' rather than X'FF'. The default value is NO.

Note: This keyword is used mainly for networks in Spain.

For X.25 ODLC, the only valid value for the SPNQLLC keyword is NO.

[,STATION={DCE}]
 {DTE}

NPSI only: Specifies whether the MCH is to operate as a DTE or a DCE. DTE is the default value and must be used for a normal connection to a network node (usually a DCE). Full DCE support is not provided by NPSI.

When STATION=DCE is coded, NDRETRY=1 should be coded so that TPTIMER corresponds to CCITT T1 and NPRETRY corresponds to N2.

V3R6 and Later: If you code INTWLINK=DIALDTE, STATION=DTE is required.

[,STATION=(lapb role[,plp role])]

X.25 ODLC Only: You can specify different roles for the LAPB and the PLP layers:

lapb role can be either DTE (the default) or DCE.

plp role can be either DTE (the default) or DCE.

The old format of this keyword is still supported, but if you use the old format and only code the first parameter, X.25 ODLC uses the value that was specified for the first parameter as the value for the second parameter.

[,STATOPT=*text string*]

V3R4 -V3R8: Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.

[,STATOPT={*text string* }]
 {ALL,*text string*}
 {GROUP,*text string*,...LU,*text string*}

Starting with V3R9: This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the STATOPT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.MCH statement.

If **ALL**,*text string* is coded, then the STATOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.MCH statement.

If the STATOPT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the "targ_name,value" group. For example, to code STATOPT on the PU and LU statements generated by the X25.MCH statement, code

STATOPT=(PU,XXXX,LU,YYYY)

In the generated NCP statements, a comment keyword of

```
*          STATOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*          STATOPT=YYYY
```

will be generated following the LU statement.

```
[,SUBADDR={YES}]
      {NO }
```

Specifies whether subaddressing is used within the Incoming Call packets on this MCH to select LLC types. The SUBADDR keyword is valid only when GATE=GENERAL is coded. Subaddressing applies only to SVCs and is specified in the rightmost digit of the called DTE address within the Incoming Call packet. The default value is NO.

```
[,SUBD=(n1,n2,n3,...nn)]
```

Specifies that fast connect or FTPI is used on this physical circuit and that the CTCP is selected through subaddressing. When SUBD is coded, the CTCP keyword must be coded on the X25.MCH statement. For Fast Connect, the following optional keywords can be coded:

- LUNAME
- LOGAPPL
- MODETAB
- VMODTAB

There is one unique subvalue for each of the subvalues of the CTCP keyword. NPSI does not generate a default value for this keyword.

V3R4 and Later:

For releases V3R4 and later, if you code 1-digit values on the SUBD parameter, the last digit of the called DTE address is used to select the CTCP. If you code 2-digit values on the SUBD parameter, the last 2 digits of the called DTE address is used to select the CTCP. Mixing of 1- and 2-digit values on the SUBD parameter is not allowed.

Note: When subaddressing is used for the CTCP selection with fast connect, and the subaddressing digit within the Called DTE address do not match any values coded on the SUBD keyword, the call is cleared. The CUD0 field cannot be used as an alternative connection method when using fast connect and subaddressing. See “SVC Selection” on page 4-24 for examples and more information.

```
[,SVCINN={number}]
      {0 }
```

Specifies the number of simultaneously active switched subarea node connections. It is representative of the number of switched virtual circuits allocated to subarea node traffic.

The default value is SVCINN=0, specifying that no SVCSC is available.

Usage Notes: The SVCINN specification must equal the number of X25.LINE statements defined with DSTNODE=INN and TYPE=S. You must code an X25.VC statement with the number of logical channel numbers (LCN=) equal to or greater than the number specified by the SVCINN parameter.

SVCINN is not valid if you code GATE=DEDICATE on the same X25.MCH statement.

```
[,T1TIMER={time}]
  {0 }
```

Specifies the time, in seconds, that the DCE waits for frame acknowledgment. Tenths of seconds can be entered. The default value of 0 does not allow for link-level piggybacking. Valid values other than 0 range from 0.4–25.5 (V3R4) or from 0.5–25.5 (V3R5 and later).

Note: This keyword corresponds to the T1 that is provided by the network vendor at subscription time. The T1TIMER keyword is used with the SPEED keyword to cause piggybacking of acknowledgments at the link level.

```
[,TAXUNIT={0 } ]
  {32 }
  {64 }
  {128 }
```

Specifies the number of characters for a network tax unit. Transmitted and received data, RR, and RNR packets are counted for statistics purposes. If TAXUNIT=0 is specified, the packets are counted. Otherwise, the taxation units are counted. In all cases, the statistics are shipped to the host in a RECFMS type 3 format either at the end of the connection or at counter overflow.

In addition, for the fast connect and FTPI functions only, TAXUNIT is used to count the transmitted and received taxation units, which are reported to the CTCP with other billing information in the CLEAR and CLEAR CONFIRMATION commands and information reports. If TAXUNIT=0 is specified, no billing information is reported to the CTCP.

The default value is 0, except for the fast connect and FTPI functions. The default value for the fast connect and FTPI functions is 64.

```
[,T3={YES,timer}]
  {NO }
```

V3R4 and Later: T3 specifies, for a DCE station in DATA phase, the inactivity polling period in seconds. The valid range is 10–720. T3 is valid only if STATION=DCE is coded on the X25.MCH statement. The T3 value must be greater than the *Tp* timer specified on the X25.MCH statement. The default is NO. In this case the inactivity polling is not performed on the MCH statement.

```
[,T4={timer}]
  {60 }
```

T4 specifies the inactivity polling period in seconds for a DTE station in DATA phase. The valid range is 10–255. The default value is 60 seconds. T4 is valid only if STATION=DTE is specified on the X25.MCH statement.

```
[,TDTIMER=timer]
```

Specifies the value of the NPSI internal delay timer between *Nd* transmissions. Valid values are 1, 2, and 3 seconds; the default value is 1 second. See the description of the TPTIMER keyword for more information.

For X.25 ODLC, the TDTIMER keyword is ignored.

```
[,TPTIMER=timer]
```

Specifies the value of the X.25 T1 timer in seconds. The valid range is from 0.5–25.5 seconds in increments of 0.1 second; the default value is 1.0 second. The X.25 T1 timer represents the maximum delay during which the DTE expects the acknowledgment of a transmitted I or U frame with or without the poll bit set, or of a transmitted S frame with the poll bit set.

Note: Once flags are transmitted and monitored on the first transmit operation issued to the scanner, the $Nd \times (Td + (Np \times Tp))$ value must exceed the time needed by the PSDN to return the flags at contact time. This time can exceed 16 seconds.

[,TRAN={**ODD** }]
 {**EVEN** }
 {**NO** }
 {**MARK** }
 {**SPACE** }
 {**USER1** }
 {**USER2** }

Specifies whether data incoming from a PAD-supported DTE or LLC type 0-supported DTE under DATE is translated from ASCII to EBCDIC. In addition, it specifies whether data going to a PAD-supported DTE or to a LLC type 0-supported DTE under DATE is translated from EBCDIC to ASCII.

- ODD** Specifies that odd parity is used for translation.
- EVEN** Specifies that even parity is used for translation.
- NO** Specifies that no translation is performed. The default value is **NO**.
- MARK** Specifies that bit 0 is always a 1, with no parity.
- SPACE** Specifies that bit 0 is always a 0, with no parity.
- USER1** Specifies two 256 byte user-defined translation tables as defined by the X25.TRAN statement with the **USER1** keyword. One table for inbound flow, and one table for outbound flow.
- USER2** Specifies two 256 byte user-defined translation tables as defined by the X25.TRAN statement with the **USER2** keyword. One table for inbound flow, and one table for outbound flow.

Notes:

1. This keyword is valid only if you code **PAD=INTEG**, **PAD=TRANSP**, or **GATE=DEDICAT**. For more information, see *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning*.
2. Only data packets are translated.

[,TRANSFR=*count*]

X.25 ODLIC and NPSI V3R5 and Later: Specifies the number of NCP buffers corresponding to the maximum amount of data that can be received from the line during a single data transfer operation. This keyword applies only to virtual circuits working in LLC type 3.

Valid values range from 1–254. The count must be less than or equal to the **TRANSFR** count coded or defaulted to in the **BUILD** statement. The checking of inbound data against the **X25.MAXPIU** value on the **BUILD** statement is replaced by the checking of the value of **TRANSFR** times the NCP buffer size (**BFRS** on the **BUILD** statement) minus the offset to the received data in the first buffer (38).

[,VCCINDX=*index*]

Specifies an entry in the virtual circuit parameter table (**VCCPT**) that all PVCs or SVCs on this MCH will use.

For SVCs, do not code the **VCCINDX** value on either the **X25.LINE** statement or the **X25.VC** statement. If **VCCINDX** is coded on either statement it must be the same as what is coded on the **X25.MCH** statement.

For PVCs, if you code **VCCINDX** on the **X25.LINE** or **X25.VC** statement, it will override the value on the **X25.MCH** statement.

For more information, see **VCCINDX** under the **X25.LINE** statement or **X25.VC** statement.

X25.MCH

[,VCID={YES}]
{NO }

Specifies the type of connection identifier (reference) that NPSI sends in commands to GATE or DATE CTCPs.

If VCID=NO is specified, the reference is the *range* of the SNA resource (always the same at each connection of a given SNA resource).

If VCID=YES is specified, the reference is the LCN of the X.25 SVC associated with the SNA resource (probably different at each connection).

VCID is authorized only if GATE is coded. The default value is NO.

[,VMODTAB=(v0,v1,v2,...vm)]

Specifies the mode table (MODETAB) for the virtual circuit LUs defined in the X25.FCG or X25.VC statements corresponding to this X25.MCH statement. There is one unique subvalue for each of the subvalues of the CTCP keyword. There is no default for this keyword.

VMODTAB is only valid if FTPI=NO is coded on the X25.MCH statement.

[,VTMKYWRD=(targ_name,kywrd_name,kywrd_value)]

Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrd_name is the name of the keyword to be generated.

kywrd_value is the value to be coded as an operand for the keyword.

[,XMONLNK={YES}]
{NO }

Specifies whether the MCH is to be automatically activated after the initial loading or after ANS. This allows NCP/NPSI to be activated using a subarea node PVC.

XMONLNK=YES must be coded when this MCH contains a subarea node PVC that can be used to activate NCP/NPSI. When ownership is lost over a channel, an X.25 PVC, or an SDLC line, XMONLNK=YES must be coded if a subarea node SVC of this MCH is to convey ownership of the NCP after ANS.

You should specify XMONLNK=YES if the MCH contains a subarea PVC or SVC. This allows ownership recovery to occur over one of the subarea node PVCs or SVCs at ANS. It also allows the MCH to stay up when a VARY NET,INACT,F command for the NCP is issued enabling subarea node traffic to continue over subarea node virtual circuits of this MCH.

This keyword replaces the X25.MNLNK statement. See the *NCP, SSP, and EP Resource Definition Reference* for more details about XMONLNK. The default value is NO.

[,X21INACT={time}]

Specifies time in seconds after which an X.21 connection is cleared if no virtual circuit activity is detected on an MCH. This keyword is valid only if X21SW=YES has been specified. Time is specified in seconds and the valid range is 1–255. The default is 30 seconds.


```
[,X21NTWK={NO  }]
          {OLD  }
          {1984}
```

Specifies whether this MCH is connected to an X.25 network offering an X.21 interface. Coding X21NTWK=OLD or X21NTWK=1984 indicates that it is an X.21 connection. The default is NO. This keyword is mandatory when X21SW=YES is specified.

The X21NTWK keyword must be coded for a 3745 (with or without a 3746 Model 900 attached). For a 3720, this keyword is not required, even if the X.25 network offers an X.21 interface.

Note: Even if coded on the X25.MCH statement, X21NTWK always defaults to NO on the corresponding GROUP statement.

```
[,X21RTYCT={number}]
```

Specifies the number of retries that are performed by the scanner when an X.21 connection is in progress. The keyword is valid only if X21SW=YES has been specified. Valid values range from 0 to 254 and the default is 3.

```
[,X21RTYTO={time}]
```

Specifies the number of seconds for the timeout interval that the scanner waits between retries when an X.21 connection is in progress. Valid values range from 0 to 1632, with increments of 0.1 seconds. The default value is 3 seconds. The keyword is valid only if X21SW=YES has been specified.

```
[,X21SW={YES  }]
          {NO  }
          {T70DXE}
```

Specifies that the physical line is an X.21 switched line. The default is NO.

V3R7 and Later: When you specify X21SW=T70DXE, the value of the STATION keyword (specified or defaulted) is overlapped at MCH connection time. In accordance with T70 CCITT recommendations, for this MCH line the X.25 link level role is set at X.21 switched connection time. In accordance with ISO8208 recommendations, the X.25 packet level role is set at Restart packets exchange time. When you specify X21SW=T70DXE and STATION is omitted, the T4 inactivity timer defaults to 60 seconds.

When X21SW=YES or X21SW=T70DXE is specified, X21NTWK is mandatory and the following restrictions apply to X25.MCH keywords:

- GATE=GENERAL
- LLCLIST=LLC4
- CONNECT=NO
- DIRECT=NO
- SPEED=xxxx

Note: When you specify X21SW=YES or X21SW=T70DXE on an MCH, you can define only switched virtual circuits on that MCH.

V3R6 and Later: For X.25/ISDN support, when you code INTWLINK=DIALDTE or DIALPSDN on the X25.MCH statement, X21SW=NO is required.

X25.NET Statement

Usage: Use this statement to specify and start the description of a PSDN. One X25.NET statement is required for each PSDN. You can specify a maximum of nine X25.NET statements during a generation.

X.25 ODLC and NPSI can be connected to the same PSDN, and therefore, share an X25.NET definition. They can also share X25.VCCPT and X25.OUFT entries.

Position: This is the first statement in the X.25 definition list. Additional X25.NET statements can be used to define additional networks.

Format and Keywords: The format of the X25.NET statement is:

symbol X25.NET keywords

Table 5-13 shows the keywords that you can code on the X25.NET statement.

Table 5-13 (Page 1 of 2). X.25 Keywords for X25.NET Statement

Keywords	
DM={ <u>YES</u> {NO }	
[,CAUSE={CCITT} { <u>IBM00</u> {IBM80}	
[,CCITT={1980}] {1984} {1988} { <u>1993</u> }	(X.25 ODLC)
[,CPHINDX={index}] { <u>1</u> }	
[,DCI={YES }] { <u>NO</u> } {LLC2} {LLC3}	(X.25 ODLC and NPSI V3R6 and later)
[,INACTO={time}] { <u>0</u> }	
[,NETTYPE={ <u>1</u> }] {3} {4}	
[,NSTDFAC={Nsfac1,Nsfac2,Nsfac2,Nsfac3,...,Nsfacn}] { <u>NONE</u> }	
[,OUHINDX={index}] { <u>1</u> }	
[,RESETINO={ccdd,ccdd,ccdd,...}] { <u>YES</u> } {NO }	
[,RESETPVC={YES}] { <u>NO</u> }	(X.25 ODLC and NPSI V3R7 and later)
[,RESETUNB {YES}] { <u>NO</u> }	(Starting with V3R9)
[,RFAC={fac1,fac2,...}] { <u>NONE</u> }	
[,RNRTIMER={nnn}] { <u>0</u> }	(X.25 ODLC and NPSI V3R7 and later)
[,R20={count}] { <u>1</u> }	
[,R22={count}] { <u>1</u> }	

Table 5-13 (Page 2 of 2). X.25 Keywords for X25.NET Statement

Keywords

[,R23={count}]	
{ <u>1</u> }	
[,T20={timer}]	
{ <u>180</u> }	
[,T21={timer}]	
{ <u>200</u> }	
[,T22={timer}]	
{ <u>180</u> }	
[,T23={timer}]	
{ <u>180</u> }	
[,T25={timer}]	(X.25 ODLC and NPSI V3R4 and later)
{ <u>0</u> }	
[,T26={timer}]	
{ <u>180</u> }	

Note: The FRMLGTH, LLCLIST, MWINDOW, and PKTMODL keywords have been moved to the X25.MCH statement. The DSTNODE and LLC keywords have been moved to the X25.LINE, and the LLC keyword has been moved to the X25.VC statement.

symbol

DM={YES}
 {**NO** }

Specifies an optional name for this X25.NET statement.

Specifies whether the LAPB DM command is received or sent by the PSDN. This command, included in the CCITT Recommendation, is not supported by all PSDNs.

V3R3 Only: There is no default for this keyword.

V3R4 and Later: The default value is YES.

For X.25 ODLC, the only valid value for the DM keyword is YES.

[,CAUSE={**CCITT**}]
 {**IBM00**}
 {**IBM80**}

Determines the type of cause and diagnostic byte that NPSI sends to the network in Clear, Reset, and Restart Request packets. The cause and diagnostic code specified in the CAUSE keyword references one of three cause and diagnostic tables. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for table specifications.

CCITT Is either cause 00 with CCITT diagnostics, or cause 80 with SNA diagnostics.

IBM00 Is cause 00 with SNA diagnostics (for IBM migration). It is the default.

IBM80 Is cause 80 with SNA diagnostics (same diagnostics as IBM00).

See “Compliance with ISO Standards and CCITT 1984 (V3 Enhancements)” on page 3-9 for more information. For X.25 ODLC, CCITT is the only valid value.

X25.NET

[,CCITT={1980}]
 {1984}
 {1988}
 {1993}

X.25 ODLC:

Specifies the X.25 version to be used for the X.25 network attachment. The default is 1993.

[,CPHINDX={*index*}]
 {1 }

Specifies the greatest index value coded in the INDEX keyword of an X25.VCCPT statement within this network. Valid values range from 1–99; the default is 1.

[,DCI={YES }]
 {NO }
 {LLC2}
 {LLC3}

V3R6 and Later: Specifies whether the Diagnostic Code Indicator (DCI) is set in the protocol identifier for X.25 connections for SNA-to-SNA DTEs. If DCI=YES, the DCI bit in the protocol identifier in call requests is set for PSH and QLLC. If DCI=NO, the DCI bit in the protocol identifier in call requests is not set for PSH and QLLC. If DCI=LLC2, the DCI bit is set in the protocol identifier in call requests for PSH. If DCI=LLC3, the DCI bit is set in the protocol identifier in call requests for QLLC. The default value is NO.

For X.25 ODLC, the only valid value for the DCI keyword is YES.

[,INACTO={*time*}]
 {0 }

Specifies the number of seconds an SVC waits before the line is cleared because of inactivity. The default is 0. If the default is specified, there is no monitoring on all the SVCs associated with this X25.NET statement. If a value from 1 to 59 seconds is specified, NDF issues a message and the generation is ended.

Valid non-zero values range from 60–5400.

For X.25 ODLC, the only valid value for the INACTO keyword is 0.

[,NETTYPE={1}]
 {3}
 {4}

Specifies the network type. Possible values are 1, 3, and 4; the default is 1.

For NETTYPE=1 or NETTYPE=4, when NPSI receives a Reset 01, 07, or 1D, it stops sending physical services commands to the remote DTE until it receives a Reset 00, 09, or 0F. For NETTYPE=3, using LLC type 2 or LLC type 3, when NPSI receives a Reset 01 or 1D, it stops sending physical services commands to the remote DTE, until it receives a Reset 00, 09, or 0F. For NETTYPE=3, using LLC type 0, LLC type 4, or LLC type 5, when NPSI receives a Reset 1D, it stops sending physical services commands to the remote DTE until it receives a Reset 00, 09, or 0F. For X.25 ODLC, NETTYPE=3 is only valid when using LLC type 3.

When NETTYPE=4 is specified, NPSI does not allow duplicate facilities, and does not accept Clear, Reset, or Restart packets that do not carry both cause and diagnostic bytes. Restart Confirmation packets that are too long are discarded, rather than answered with a Restart Request packet with a cause and diagnostic code corresponding to X'A7' index. If CTCP is used, see “Virtual Circuits with a CTCP” on page A-9.

[,NSTDFAC={*Nsfac1,Nsfac2,Nsfac3,Nsfac3,...,Nsfacn*}]
 {NONE }

Specifies the nonstandard facilities accepted. A table of nonstandard user facilities is built at system generation. Nonstandard facilities must be in accordance with X.25 facility format. A nonstandard facility can be duplicated in a Call packet. Therefore, the nonstandard facility must be coded as *many times* as it might be duplicated. The default value is NONE.

The following standard facilities are accepted without special coding:

01	High-priority class of traffic, fast select, or reverse charging
02	Throughput class negotiation
03	Closed user group—basic
08	Called line address modified (V3R4 and later)
09	Closed user group with outgoing access—basic
41	Bilateral closed user group selection
42	Packet size negotiation
43	Window size negotiation
47	Closed user group—extended
48	Closed user group with outgoing access—extended
C3	Call redirection notification.
C9	Called address extension facility (V3R4 and later)
D2	Quality of service facility (V3R4 and later)

All other facility fields are considered to be nonstandard. Nonstandard facilities are rejected by a Clear packet containing either a X'E6' or X'41' Diagnostic code, unless coded in the NSTDFAC keyword.

Both standard and nonstandard facilities are coded in hexadecimal. You must use two hexadecimal digits to code a facility in the NSTDFAC keyword. If several facilities are to cause call accepted, they must be enclosed in parentheses. For example, you can code:

```
NSTDFAC=(05,1F,FF,FE,18)
```

This causes the calls that include these nonstandard facilities to be accepted.

Note: When a standard facility code can appear several times in a Call packet, it must also be coded in NSTDFAC. You can code up to 80 facilities.

For X.25 ODLC, the only valid value for the NSTDFAC keyword is NONE.

[,OUHINDX={*index*}]
 {1 }

Specifies the greatest index value coded in the INDEX keyword of an X25.OUFT statement within this network. Valid values range from 1–99; the default is 1.

[,RESETINO={*ccdd,ccdd,ccdd,...*}]
 {YES }
 {NO }

Specifies the action that is taken when receiving a Reset packet. RESETINO applies to PVCs and SVCs; YES is the default.

Use (*ccdd,ccdd,ccdd,...*) to specify a list of cause and diagnostic codes that can be present in Reset packets, but that do not cause NPSI to make the PU representing the virtual circuit inoperative.

ccdd represents four hexadecimal digits and can be coded as *ccXX* or *XXdd*.

When you code *ccXX* or *XXdd*, NPSI does not make the PU representing the virtual circuit inoperative for all the Reset packets beginning with cause *cc* or ending with diagnostic *dd*. If you specify more than one cause and one diagnostic, the codes must be contained within parentheses.

When RESETINO=(*ccdd,ccdd,ccdd,...*) up to 46 subvalues are allowed.

If RESETINO=YES and a Reset packet is sent or received, NPSI makes the PU representing the virtual circuit inoperative.

If RESETINO=NO and a Reset packet is sent, NPSI avoids making the PU representing the virtual circuit inoperative.

Notes:

1. When RESETINO=NO is coded, NPSI cannot guarantee data integrity when a reset exchange occurs, because packet sequencing is reset. In addition, this coding may prevent any further system activity on that line, until the problem is resolved.
2. In case of GATE/DATE, it is usually the CTCP responsibility to make the PU representing the virtual circuit inoperative after RESET processing.

For X.25 ODLC, the only valid value for the RESETINO keyword is YES. Receipt of a Reset packet with a diagnostic byte of X'A1' will not cause an X.25 ODLC PVC to be made inoperative.

[,RESETPVC={NO }]
{YES}

X.25 ODLC and NPSI V3R7 and Later: Specifies whether to reset all active PVCs when the MCH is reinitialized. You can code RESETPVC=YES or RESETPVC=NO. The default is RESETPVC=NO.

If you code RESETPVC=YES:

- A PVC is reset (diagnostic X'A1') whenever the associated SNA link is activated, if the MCH is already active.
- PVCs that are active at MCH initialization or reinitialization are reset with diagnostic X'A1'.

RESETPVC=YES is not authorized if GATE=DEDICATED is coded on the same X25.MCH.

If you code RESETPVC=NO, PVCs are not reset at PVC link activation or when MCH is activated or reinitialized.

For X.25 ODLC, the only valid value for the RESETPVC keyword is YES.

[,RESETUNB={YES}]
{NO }

Starting with V3R9: Specifies whether NPSI should send a RESET packet when a BIND or UNBIND is received from the application, or a data packet is received but the PCNE LU session is not bound. NO is the default.

When RESETUNB=YES is coded, MAXOUT=6 must be coded on the X25.VC or X25.PU statement that defines the PCNE PVC.

RESETUNB=YES is allowed only for LLC0 PVCs. RESETUNB=YES is not allowed when GATE=DEDICAT is specified on the X25.MCH statement.

[,RFAC={*fac1, fac2, ...*}]
 {**NONE** }

Specifies the facilities to be rejected. When an Incoming Call or Call Connected arrives carrying a facility coded in the RFAC keyword, that call is cleared. Twelve optional facilities are available and each can be rejected. The default is NONE (no facility is rejected). If several facilities are to cause call rejection, they must be enclosed in parentheses. Use the RFAC keyword, where *fac* is the mnemonic code of the facility to be rejected.

The facilities that can cause a call to be cleared due to rejected facilities are:

FASTSEL	Fast select
REVCHG	Reverse charging
THRUPUTCL	Throughput class negotiation
CALLREDIR	Call redirection
CUGB	Closed user group with basic format
CUGOAB	Closed user group with outgoing access (basic)
CUGE	Closed user group with extended format
CUGOAE	Closed user group with outgoing access (extension)
BLCUG	Bilateral closed user group.
AEF	Called address extension facility (V3R4 and later)
LAMOD	Called line address modified notification facility (V3R4 and later)
OSIPRI	OSI priority facility of QOS negotiation facilities (V3R4 and later)

For X.25 ODLC, the only valid value for the RFAC keyword is FASTSEL.

[,RNRTIMER={**0** }]
 {*nnn*}

V3R7 and Later: Specifies the delay in seconds for a received RNR packet to be compensated with an RR. You can specify a value from 0 to 200. The default value for *n* is 0.

If you code RNRTIMER with a nonzero value for *n*, a virtual circuit over which an RNR packet was received is reset (diag X'94') if no RR is received to compensate the RNR within *n* seconds.

If you code RNRTIMER=0 and no RR is received, the virtual circuit over which the RNR packet was received can remain in a holding state indefinitely.

An RNRTIMER keyword coded with a nonzero *n* is not authorized if GATE=DEDICATED is coded on X25.MCH.

For X.25 ODLC, the only valid value for the RNRTIMER keyword is 0.

[,R20={*count*}]
 {**1** }

Specifies the maximum number of Restart Request retransmissions. The default value is 1. The valid values range from 0–255.

[,R22={*count*}]
 {**1** }

Specifies the maximum number of Reset Request retransmissions. The default value is 1. The valid values range from 0–255.

X25.NET

- [,R23={*count*}]
 {1 }
Specifies the maximum number of Clear Request retransmissions. The default value is 1. The valid values range from 0–255.
- [,T20={*timer*}]
 {180 }
Specifies a value for the Restart Request Response timer. The default value is 180 seconds. The valid values range from 1–255.
- [,T21={*timer*}]
 {200 }
Specifies a value for the Call Request response timer. The default value is 200 seconds. The valid values range from 1–255.
- [,T22={*timer*}]
 {180 }
Specifies a value for the Reset Request response timer. The default value is 180 seconds. The valid values range from 1–255.
- [,T23={*timer*}]
 {180 }
Specifies a value for the Clear Request response timer. The default value is 180 seconds. The valid values range from 1–255.
- [,T25={*timer*}]
 {0 }
V3R4 and Later: Specifies a value for the window rotation timer (the data packet acknowledgment timer). The default value is 0 seconds. The valid values range from 0–255 seconds. This timer is used only for virtual circuits belonging to a DTE MCH, that is, T25=*timer* is not allowed when STATION=DCE is coded on the same X25.MCH statement.

T25 set to zero indicates that the timer is not used.

For X.25 ODLC, the only valid value for the T25 keyword is 0.
- [,T26={*timer*}]
 {180 }
Specifies a value for the Interrupt Confirmation timer. The default value is 180. The valid values range from 10–255.

For X.25 ODLC, the only valid value for the T26 keyword is 0.

X25.OUFT Statement

Usage: Use this statement to specify the user facilities and call user data that is copied into the Call Request packet for an outgoing call.

One X25.OUFT statement is required for each entry in the optional user facilities table. The number of entries in the table is specified by the OUHINDX keyword of the X25.NET statement. Consecutive entries need not be used.

Position: This statement must be placed between the X25.NET and the X25.MCH statements to which it applies.

Format and Keywords: The format of the X25.OUFT statement is:

```
symbol      X25.OUFT      keywords
```

You can code as many as 99 X25.OUFT statements for each PSDN. At least one X25.OUFT statement is required for each PSDN using SVCs. An X25.OUFT statement is not required for a PSDN using only PVCs.

Table 5-14 shows the keywords you can code on the X25.OUFT statement.

Table 5-14. X.25 Keywords for X25.OUFT Statement

Keywords	
	INDEX=index [,OPTFACL=hexvalue] [,USRFILD=hexvalue] [,USRFIL2=hexvalue]
<i>symbol</i>	Specifies an optional name for this X.25 OUFT statement.
INDEX=index	Specifies the entry in the optional user facilities table. Valid values range from 01–99.
[,OPTFACL=hexvalue]	Specifies the hexadecimal configuration to be copied into the optional facility field of a Call Request packet. The <i>hexvalue</i> must contain an even number of digits less than or equal to 218. Use the optional facility fields for flow control negotiation. See Appendix A, “Defining X.25 PSDN Facilities,” for a description of flow control negotiation.
[,USRFILD=hexvalue]	Specifies the correct bit configuration to be copied after the virtual circuit type (protocol identifier) in the CUD field of the Call Request packet. Set the <i>hexvalue</i> to an even number of digits less than or equal to 252. The virtual circuit type will be one of the following values:
	X'C0dddd0' Virtual circuit type 0
	X'C2' Virtual circuit type 2
	X'C3' Virtual circuit type 3 peripheral node
	X'E3' Virtual circuit type 3 subarea node
	X'pp000000' Virtual circuit type 5

where:

pp Is equal to 01, 41, 51, or 81. If L=7 is coded (user-defined CUD0), pp is user-defined and bytes 1, 2, and 3, of the CUD are not filled with 000000.

dddd Is the remote DTE IDNUM. (It is not added in the CUD if L=.).

[,USRFIL2=*hexvalue*]

Specifies the correct bit configuration to be copied as the last part of the CUD field of the Call Request packet (after the protocol identifier and the USRFIL2, if coded). Set the *hexvalue* with an even number of digits less than or equal to 252.

Note: The sum of the lengths of the values specified in the protocol identifier, USRFIL2, and USRFIL2 must not exceed 256 digits. The maximum length of the CUD field in a Call packet for fast select is 256 digits. A normal Call Request packet has a CUD field less than 32 digits (16 bytes).

For logical channels used exclusively by type 4 virtual circuits or physical circuits under the control of a DATE CTCP, X25.OUFT statements are not used. However, it is necessary to reference an X25.OUFT statement in the associated X25.LINE or X25.VC statements. If there is no other requirement for X25.OUFT statements, a dummy entry for these virtual circuits can be created as follows:

```
X25.OUFT INDEX=1
```

The following is an example of an X25.OUFT statement:

```
X25.OUFT
INDEX=1,
OPTFACL=0302430707,
USRFILD=123456789ABC,
USRFIL2=4455
```

Assuming this is a type 2 virtual circuit; the logical channel number is 4, the packet modulo is 8, and the corresponding Call Request packet is:

```
X'10040B091234567890050302430707C2123456789ABC4455'
```

where:

1	Is the GFID.
0	Is the logical channel group number.
04	Is the logical channel number.
0B	Is the Call Request.
09	Is the address length.
123456789	Is the DTE address.
0	Is the padding to create an even number of digits.
05	Is the facility length.
0302430707	Is the facility field.
C2	Is LLC type 2.
123456789ABC4455	Is the CUD field.

See "PATH Definition Statement" on page 6-8 for more details.

X25.PAD Statement

Usage: Use this statement to specify one or more strings of PAD parameters. These parameters are selected by an index coded on the X.25 PAD statement.

Position: This statement is optional. If specified, this statement must be coded before the first X25.NET statement of the NPSI generation.

Format and Keywords: The format for the X25.PAD statement is:

symbol X25.PAD keywords

Table 5-15 shows the keywords you can code for the X25.PAD statement.

Table 5-15. NPSI Keywords for X25.PAD Statement

Keywords

INDEX=value,
 PADPARAM={value}
 {NULL }

symbol

Specifies an optional name for this X.25 PAD statement.

INDEX=value

Specifies the entry of the PAD table that is initialized by this X25.PAD statement and referred to by the PADINDX keyword on the X25.MCH statement. The valid range is from 1–99; there is no default value.

PADPARAM={value}
{NULL }

Specifies the PAD parameters. Use the *value* to represent a string of up to 104 hexadecimal digits (string length of even parity), or code NULL for no PAD parameters. There is no default value.

X25.PU Statement

Usage: Use this statement to define the physical unit associated with either an SNA DTE connected to a corresponding virtual circuit or a non-SNA DTE connected to a corresponding virtual circuit.

Position: You must code one X25.PU statement instruction for each X25.LINE statement.

Format and Keywords: The format of the X25.PU statement is:

symbol X25.PU keywords

Table 5-16 and Table 5-17 on page 5-91 list the keywords and VTAM keywords that you can code on the X25.PU statement. The VTAM keywords provide information only to the VTAM initialization process and are not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for descriptions of VTAM keywords and for information on the VTAM initialization process.

X25.PU keywords are the same as the NCP PU keywords. The X.25 PU keywords and NCP PU keywords are valid under the X25.PU statement. However, only the X25.PU keywords are described in this section. Appendix B, "System Generation Examples" on page B-1, contains sample X25.PU statements for several virtual circuit types. For more information about these statement keywords, see the *NCP, SSP, and EP Resource Definition Reference*.

Table 5-16 (Page 1 of 2). X.25 Keywords for X25.PU Statement

Keywords

[,ADJLINK(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,ADJNODE(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,AGGPRI(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,CMTKYWRD=(targ_name,kywrld_name,kywrld_value)]	(X.25 ODLC and NPSI starting with V3R9)
[,HPRQLIM={nnn} {0 }]	(X.25 ODLC starting with NCP V7R5)
[,LLERP=REQUIRED]	(X.25 ODLC starting with NCP V7R5)
[,MAXDATA={size} {265}]	
[,MAXOUT=number]	
[,NGFINC(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,NGFOPT(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,NGFTXT(text string)]	(X.25 ODLC and NPSI starting with V3R9)
[,PUDR=NO]	
[,PUTYPE=type]	
[,STATOPT=text string]	
[,TGN=number]	
[,VTMKYWRD=(targ_name,kywrld_name,kywrld_value)]	(X.25 ODLC and NPSI starting with V3R9)

Table 5-16 (Page 2 of 2). X.25 Keywords for X25.PU Statement

Keywords

[,XID={YES}]
 {NO}

Table 5-17 (Page 1 of 2). VTAM Keywords for X25.PU Statement

Keywords

[,ADDR=...]	
[,ANS=...]	
[,ASLENT=...]	
[,ASLTAB=...]	
[,AVGPB=...]	
[,BRKCON=...]	
[,BNSUP=...]	
[,CAPACITY=...]	(X.25 ODLC and NPSI V3R6 and later)
[,CLRSESSQ=...]	(X.25 ODLC and NPSI starting with V3R9)
[,CONNTYPE=...]	(X.25 ODLC and NPSI V3R6 and later)
[,COSTBYTE=...]	(X.25 ODLC and NPSI V3R6 and later)
[,COSTTIME=...]	(X.25 ODLC and NPSI V3R6 and later)
[,CPCP=...]	(X.25 ODLC and NPSI V3R6 and later)
[,CPNAME=...]	(X.25 ODLC and NPSI V3R6 and later)
[,DATMODE=...]	
[,DISCNT=...]	
[,DLOGMOD=...]	
[,DYNADJCP=...]	(X.25 ODLC and NPSI V3R6 and later)
[,ENCR=...]	
[,FEATUR2=...]	
[,HPR=...]	(X.25 ODLC starting with NCP V7R5)
[,IRETRY=...]	
[,ISTATUS=...]	
[,LOGAPPL=...]	
[,LOGTAB=...]	
[,MAXLU=...]	
[,MDLENT=...]	
[,MDLTAB=...]	
[,MODETAB=...]	
[,NETID=...]	
[,NN=...]	(X.25 ODLC and NPSI V3R6 and later)
[,PACING=...]	
[,PASSLIM=...]	
[,PDELAY=...]	(X.25 ODLC and NPSI V3R6 and later)
[,RETRIES=...]	
[,SECLEVEL=...]	(X.25 ODLC and NPSI V3R5 and earlier)
[,SECTNET=...]	

Table 5-17 (Page 2 of 2). VTAM Keywords for X25.PU Statement

Keywords	
[,SECURITY=...]	(X.25 ODLC and NPSI V3R6 and later)
[,SPAN=...]	
[,SSCPFM=...]	
[,SUBAREA=...]	
[,TERM=...]	
[,TGP=...]	(X.25 ODLC and NPSI V3R6 and later)
[,TOPO=...]	(X.25 ODLC and NPSI V3R6 and later)
[,UPARM1=...]	(X.25 ODLC and NPSI V3R6 and later)
[,UPARM2=...]	(X.25 ODLC and NPSI V3R6 and later)
[,UPARM3=...]	(X.25 ODLC and NPSI V3R6 and later)
[,USSTAB=...]	
[,VERALSID=...]	(X.25 ODLC and NPSI V3R6 and later)
[,VPACING=...]	

<i>symbol</i>	Specifies an optional resource name for the physical unit. The default name is built as follows: pPxxxggg See "Resource Naming Conventions" on page 5-2 for information on default naming conventions.
[,ADJLINK= <i>text string</i>]	This keyword will be generated as a comment following the NCP PU statement, to be processed by NetView.
[,ADJNODE= <i>text string</i>]	This keyword will be generated as a comment following the NCP PU statement, to be processed by NetView.
[,AGGPRI= <i>text string</i>]	This keyword will be generated as a comment following the NCP PU statement, to be processed by NetView.
[,CMTKYWRD=(<i>targ_name,kywrд_name,kywrд_value</i>)]	Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement. <i>targ_name</i> is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL. <i>kywrд_name</i> is the name of the keyword to be generated. <i>kywrд_value</i> is the value to be coded as an operand for the keyword.
[,HPRQLIM={ <i>nnn</i> } {0 }]	X.25 ODLC and NCP V7R5 and later: Specifies the maximum number of APPN HPR data bytes that can be queued to a peripheral link station's transmission queue at one time. If an APPN HPR frame arrives at a link station which has a transmission queue that is already at or above this limit, the frame is discarded. The default is 0, which means that there is no limit to the amount of APPN HPR data that can be queued. The maximum is 1677215.

Notes:

1. This keyword is valid for ODLC X.25 stations only.
2. This keyword is supported only when PUTYPE=2 and XID=YES.
3. If HPRQLIM is not coded on the X25.PU or X25.VC statement, and it is coded on the BUILD statement, then the value on the BUILD statement is used.

<code>[,LLERP=REQUIRED]</code>	X.25 ODLC and NCP V7R5 and later: Specifies whether the link station will perform link level error recovery procedures when sending HPR data on this link. LLERP=REQUIRED is the NCP default and is required for running HPR over X.25 ODLC stations.
<code>[,MAXDATA={size}]</code> <code> {265 }</code>	Specifies the maximum bytes in a PIU. Because segmenting is not supported for virtual circuit types 0, 4, and 5, MAXDATA must exceed the largest PIU size for these virtual circuit types. This keyword does not need to be coded for SNA type 2.1, because its value is determined during XID 3 negotiation. See the <i>NCP, SSP, and EP Resource Definition Reference</i> for calculation and more information about MAXDATA.
<code>[,MAXOUT=number]</code>	Code MAXOUT=6 to queue data arriving before start data traffic (SDT) in NPSI for LLC type 0 and LLC type 5. The default value for MAXOUT is 6 for LLC type 4 PVCs. When the value of MAXOUT is not equal to six, the first data packet received is sent to the SSCP as a logon message. Note: The MAXOUT keyword applies to permanent virtual circuits only.
<code>[,NGFINC=text string]</code>	Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the PU statement generated by NDF.
<code>[,NGFOPT=text string]</code>	Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the PU statement generated by NDF.
<code>[,NGFTXT=text string]</code>	Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the PU statement generated by NDF.
<code>[,PUDR=NO]</code>	Specifies whether dynamic reconfiguration is supported on this PU. However, X.25 does not support dynamic reconfiguration, so the value is forced to NO.
<code>[,PUTYPE=type]</code>	Specifies the physical unit type associated with the remote DTE. Specify 1 for LLC type 0, LLC type 4, and LLC type 5.
<code>[,STATOPT=text string]</code>	Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the PU statement generated by NDF.
<code>[,TGN=number]</code>	Transmission group number. Only single link transmission groups are supported by X.25. Therefore, the number cannot match any other TGN specification.
<code>[,VTMKYWRD=(targ_name,kywrд_name,kywrд_value)]</code>	Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.

X25.PU

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrđ_name is the name of the keyword to be generated.

kywrđ_value is the value to be coded as an operand for the keyword.

[,XID={YES}]
{NO }

Specifies whether an XID exchange will occur with the remote DTE PVC. NPSI forces a NO for LLC types 0, 2, 4, and 5, because the remote device is unable to support an XID exchange on a PVC. For LLC type 3, you should specify YES or NO. For SNA type 2.1 nodes, YES must be coded.

V3R3 Only: For NPSI V3R3, the default is NO.

See the *NCP, SSP, and EP Resource Definition Reference* for more information.

X25.TRAN Statement

Usage: Use this statement to specify user-coded translation tables. This translation table option applies to integrated and transparent PAD and to PCNE under DATE.

The X25.TRAN statement is referenced with the USER1 and USER2 values of the TRAN keyword on the X25.MCH statement. Two tables can be specified. If the TRAN keyword on the associated X25.MCH statement specifies USER1, the table identified by TRAN=USER1 is used for the translation of data. If the TRAN keyword of the associated X25.MCH statement specifies USER2, the table identified by TRAN=USER2 is used for the translation of data.

Position: This statement is optional. If specified, this statement must be coded before the first X25.NET statement of NPSI generation. Only two X25.TRAN statements can be coded in one NPSI generation.

Format and Keywords: The format for the X25.TRAN statement is:

symbol X25.TRAN keywords

Table 5-18 shows the keywords that you can code on the X25.TRAN statement.

Table 5-18. NPSI Keywords for X25.TRAN Statement

Keywords

USER=value,
 DCIN0=value,
 DCIN1=value,
 DCIN2=value,
 .
 .
 .
 DCINF=value,
 DCOT0=value,
 DCOT1=value,
 DCOT2=value,
 .
 .
 .
 DCOTF=value,

symbol

Specifies an optional name for this X25.TRAN statement.

USER=value,

Specifies the user translate tables. Two user translate tables can be generated for NPSI generation. The value can be either 1 or 2. There is no default value for this keyword.

DCINx=value,

Specifies the user translate tables definition for inbound flow (256 bytes). The character x in the keyword name represents a hexadecimal digit from 0–F. Sixteen keywords are required to complete the definition of a user table. Each keyword value is a 32-digit hexadecimal string and there is no default value.

X25.TRAN

DCOT*x=**value*,

Specifies the user translate tables definition for outbound flow (256 bytes). The character *x* in the keyword name represents a hexadecimal digit from 0–F. Sixteen keywords are required to complete the definition of a user table. Each keyword value is a 32-digit hexadecimal string and there is no default value.

X25.VC Statement

Usage: Use this statement to describe one or more virtual circuits with the same characteristics.

Position: The X25.VC statement follows the X25.MCH and X25.LCG statements, and precedes the X25.FCG statement when a fast connect MCH is defined.

Format and Keywords: The format of the X25.VC statement is:

```
symbol      X25.VC      keywords
```

Table 5-19 and Table 5-20 on page 5-98 list the keywords and VTAM keywords that you can code on the X25.VC statement. The VTAM keywords provide information only to the VTAM initialization process and are not required in the generation definition used as input to NCP generation. See the appropriate VTAM installation book for descriptions of VTAM keywords and for information on the VTAM initialization process.

Table 5-19 (Page 1 of 2). X.25 Keywords for X25.VC Statement

Keyword	
LCN=(number1,number2),	
TYPE={SWITCHED S },	
{PERMANENT P },	
VCCINDX=index	
[,ADJLINK={text string	}] (X.25 ODLC and NPSI starting
{(ALL,text string)	} with V3R9)
{(GROUP,text string,...LU,text string)}	
[,ADJNODE={text string	}] (X.25 ODLC and NPSI starting
{(ALL,text string)	} with V3R9)
{(GROUP,text string,...LU,text string)}	
[,AGGPRI={text string	}] (X.25 ODLC and NPSI starting
{(ALL,text string)	} with V3R9)
{(GROUP,text string,...LU,text string)}	
[,ANS={CONTINUE CONT}]	
{STOP	}
[,CALL={IN	}]
{OUT	}
{INOUT}	
[,CLRSESSQ={PU,key_value	}] (X.25 ODLC and NPSI starting
{LU,key_value	} with V3R9)
{LU,key_value,PU,key_value}	
[,CMTKYWRD=(targ_name,kywrd_name,kywrd_value)]	(X.25 ODLC and NPSI starting
	with V3R9)
[,COMMITO={1	}]
{2	}
{4	}
{8	}
{NO}	
[,CTCPNO=number]	(V3R4 and later)
[,DEFPRI={NO	}] (V3R4 and later)
{LOW	}
{NORMAL	}
{HIGH1	}
{HIGH2	}
[,HEXNAME={YES}]	
{NO	}
[,HPRQLIM={nnn}]	(X.25 ODLC starting with NCP
{0	} V7R5)

Table 5-19 (Page 2 of 2). X.25 Keywords for X25.VC Statement

Keyword	
[,ID={bbbbffff}]	(X.25 ODLC and NPSI V3R4 and later)
[,IDNUMT={ffff}]	IDNUMT must be an even number
[,ISTATUS={ACTIVE }] {INACTIVE}	
[,LLC={LLC0}] {LLC4} {LLC5}	
[,LLERP=REQUIRED]	(X.25 ODLC starting with NCP V7R5)
[,MAXDATA={number}] {265 }	
[,MAXOUT=number]	
[,NCPGRP={label}] {NEW }	
[,NGFINC={text string }] {(ALL,text string)} {(GROUP,text string,...LU,text string)}	(X.25 ODLC and NPSI starting with V3R9)
[,NGFOPT={text string }] {(ALL,text string)} {(GROUP,text string,...LU,text string)}	(X.25 ODLC and NPSI starting with V3R9)
[,NGFTXT={text string }] {(ALL,text string)} {(GROUP,text string,...LU,text string)}	(X.25 ODLC and NPSI starting with V3R9)
[,OSITYPE={number}] {0 }	
[,OUFINDX=index]	
[,PRFLINE=x]	
[,PRFLU=z]	
[,PRFPU=y]	
[,RESETUNB {YES}] {NO }	(Starting with V3R9)
[,RETVCCCT={count}] {3 }	
[,RETVCTO={timeout}] {30 }	
[,SPAN=name]	
[,STATOPT={text string }] {(ALL,text string)} {(GROUP,text string,...LU,text string)}	V3R4 - V3R8 (X.25 ODLC and NPSI starting with V3R9)
[,SUFFIX=w]	
[,VTMKYWRD=(targ_name,kywrld_name,kywrld_value)]	(X.25 ODLC and NPSI starting with V3R9)

Table 5-20 (Page 1 of 2). VTAM Keywords for X25.VC Statement

Keyword	
[,ASLENT=...]	
[,ASLTAB=...]	
[,DISCNT=...]	
[,CAPACITY=...]	(V3R6 and later)
[,CONNTYPE=...]	(V3R6 and later)
[,COSTBYTE=...]	(V3R6 and later)
[,COSTTIME=...]	(V3R6 and later)
[,CPCP=...]	(V3R6 and later)
[,CPNAME=...]	(V3R6 and later)

Table 5-20 (Page 2 of 2). VTAM Keywords for X25.VC Statement

Keyword	
[,DLOGMOD=...]	
[,DYNADJCP=...]	(V3R6 and later)
[,DYNPU=...]	(V3R6 and later)
[,DYNPUPFX=...]	(V3R6 and later)
[,ENCR=...]	
[,FEATUR2=...]	
[,HPR=...]	(X.25 ODLC starting with NCP V7R5)
[,LOGAPPL=...]	
[,LOGTAB=...]	
[,MDLENT=...]	
[,MDLTAB=...]	
[,MODETAB=...]	
[,NN=...]	(V3R6 and later)
[,OWNER=...]	
[,PDELAY=...]	(V3R6 and later)
[,SECLEVEL=...]	(V3R4 and V3R5)
[,SECURITY=...]	(V3R6 and later)
[,SSCPFM=...]	
[,TERM=...]	
[,TGN=...]	(V3R6 and later)
[,TGP=...]	(V3R6 and later)
[,TOPO=...]	(V3R6 and later)
[,UPARM1=...]	(V3R6 and later)
[,UPARM2=...]	(V3R6 and later)
[,UPARM3=...]	(V3R6 and later)
[,USSTAB=...]	
[,VERALSID=...]	(V3R6 and later)
[,VPACING=...]	

symbol

Specifies an optional label for this X25.VC statement.

Note: The *symbol* field is used for editing purposes only, and is neither referenced nor used by NDF.

LCN=(*number1*,*number2*)

Specifies that the virtual circuits defined by this X25.VC statement have sequential logical channel numbers, beginning with *number1* and ending with *number2*. These two numbers are in decimal notation. Valid values range from 0–255; however, some PSDNs do not use the value 0 on logical group 0. Specify only one value (LCN=*number1*) when you define only one logical channel at a time.

For X.25 ODLC, logical channels must be defined according to CCITT (in other words, in four groups of consecutively numbered LCNs). These groups must be in the following order:

1. PVCs
2. SVCs with CALL=IN
3. SVCs with CALL=INOUT
4. SVCs with CALL=OUT

There may be gaps in LCNs between groups.

TYPE={SWITCHED|S }
{PERMANENT|P }

Specifies the type of virtual circuits being defined. You can abbreviate SWITCHED with an S and PERMANENT with a P. If the MCH is dedicated to Fast Connect, code TYPE=S. This keyword is required and has no default.

VCCINDX=*index*

Specifies the entry in the virtual circuit connection parameter table (VCCPT) that contains operational characteristics of the defined PVCs.

For SVCs, this keyword specifies:

- Incoming call—the entry in the VCCPT that contains the operational characteristics of the defined virtual circuits.
- Outgoing call (Call Request)—the default value of the entry in the VCCPT that contains the operational characteristics of the defined virtual circuits. The default value is used when the “XX” field of the DIALNO or DLCADDR keyword of the VTAM PATH statement equals 00.

These values can be overridden during flow control negotiation.

The specified table entry must be defined in an X25.VCCPT statement. Valid values range from 1–99.

Packet and window sizes defined in the corresponding VCCPT entry must correspond to the PSDN-assigned values.

When VCCINDX is coded for SVCs, the value must be the same on all of the X25.VC statements for this MCH.

If VCCINDX is specified on the X25.MCH statement, the value of VCCINDX from the X25.MCH statement will be used if not coded here.

If this is a PVC and VCCINDX is coded on both the X25.MCH and X25.VC statement, the value coded on the X25.VC statement will override the value specified on the X25.MCH statement.

If this is an SVC, the value must match what is coded on the X25.MCH statement if it is coded here.

```
[,ADJLINK={text string      }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJLINK keyword and its text string are copied to the LINE statement generated by the X25.VC statement.

If **ALL**,*text string* is coded, then the ADJLINK keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.

If the ADJLINK keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJLINK on the PU and LU statements generated by the X25.VC statement, code

```
ADJLINK=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJLINK=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJLINK=YYYY
```

will be generated following the LU statement.

```
[,ADJNODE={text string    }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the ADJNODE keyword and its text string are copied to the PU statement generated by the X25.VC statement.

If **ALL**,*text string* is coded, then the ADJNODE keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.

If the ADJNODE keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code ADJNODE on the PU and LU statements generated by the X25.VC statement, code

```
ADJNODE=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* ADJNODE=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* ADJNODE=YYYY
```

will be generated following the LU statement.

```
[,AGGPRI={text string    }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the AGGPRI keyword and its text string are copied to the LINE statement generated by the X25.VC statement.

If **ALL**,*text string* is coded, then the AGGPRI keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.

If the AGGPRI keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code AGGPRI on the PU and LU statements generated by the X25.VC statement, code

```
AGGPRI=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
* AGGPRI=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
* AGGPRI=YYYY
```

will be generated following the LU statement.

```
[,ANS={CONTINUE|CONT}]
      {STOP      }
```

Specifies whether the PU connected to this virtual circuit continues or stops if NCP enters ANS. This is true only if ANS=CONT is coded on the X25.MCH statement. The value is passed to the NCP PU statement. X.25 supports session continuation on virtual circuits for SNA devices.

The ANS keyword is meaningful only for PVC. For SVC, the ANS keyword must be coded in the PU (switched) definition statement of the VTAM Switched Major Node. See *VTAM Installation and Resource Definition* for more information.

Session continuation with virtual circuits supporting non-SNA connections is also supported. See the *NCP, SSP, and EP Resource Definition Reference* for the meaning and details of the ANS keyword. No default is defined for this keyword; the NCP default value is used.

```
[,CALL={IN  }]
      {OUT  }
      {INOUT}
```

Specifies the way an SVC is established. You can use this keyword only if TYPE=S is coded.

Use CALL=IN if the virtual circuits you are defining are established at the request of a remote DTE. Do not specify CALL=IN on any virtual circuit definition following an X25.MCH statement that specifies a value greater than 0 for the SVCINN keyword. CALL=INOUT means that virtual circuits can be established in either way.

Use CALL=OUT if the virtual circuits you are defining are established at the request of the local host access method.

For a DATE physical circuit, you must code the CALL keyword to match your PSDN subscription.

For a GATE physical circuit, you must code CALL=IN or CALL=INOUT. If the virtual circuits are set through incoming calls, you must specify CALL=IN. To allow outgoing calls, you must specify CALL=INOUT. However, if you want to reserve a certain number of virtual circuits for non-LLC type 4 call-outs, code CALL=OUT for the number of virtual circuits to be reserved.

```
[,CLRSESSQ={PU,key_value      }]
            {LU,key_value      }
            {LU,key_value,PU,key_value}
```

This keyword and its *key_value* will be generated as a keyword on the generated PU statement or LU statement or both, to be processed by VTAM.

```
[,CMTKYWRD=(targ_name,kywrld_name,kywrld_value)      ]
```

Specifies any keyword that is not known to NPSI but needs to be generated as a comment following the NCP statement.

targ_name is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

kywrld_name is the name of the keyword to be generated.

kywrld_value is the value to be coded as an operand for the keyword.


```
[,COMMITO={1 }]  
      {2 }  
      {4 }  
      {8 }  
      {NO}
```

Specifies the value in seconds of the commit timeout for idle virtual circuits. When this timer elapses, buffers committed to these virtual circuits are decommitted. The default value is 4 seconds. If you do not want a commit time out value, specify NO.

For X.25 ODLC, the COMMITO keyword is ignored.

```
[,CTCPNO=number]
```

V3R4 and Later: Specifies, in decimal, the FTPI CTCP with which this virtual circuit communicates. This number is the CTCPNO value coded on the X25.CPL statement of the selected CTCP. This parameter applies only to PVCs belonging to an FTPI MCH, and must be used when there are several FTPI CTCPs. CTCPNO can be omitted when only one X25.CPL is coded.

```
[,DEFPRI={NO }]  
      {LOW }  
      {NORMAL}  
      {HIGH1 }  
      {HIGH2 }
```

V3R4 and Later: Specifies the QOS priority value for the GATE PVC. It is valid on the X25.VC statement only when TYPE=P and LLC=4 are coded. The default value is the value of the DEFPRI keyword coded on the X25.MCH statement.

Note: If DEFPRI=LOW|NORMAL|HIGH1|HIGH2 is coded on the X25.MCH statement, you cannot code DEFPRI=NO on the X25.VC statement. If DEFPRI=NO is coded on the X25.MCH statement (or DEFPRI is not coded on the MCH), you cannot code DEFPRI=LOW|NORMAL|HIGH1|HIGH2 on the X25.VC statement.

```
[,HEXNAME={YES}]  
      {NO }
```

HEXNAME is used with the PRFLINE, PRFPU, PRFLU, and SUFFIX keywords. When creating resource names by concatenating the PRFLINE, PRFPU, PRFLU names and the SUFFIX values, HEXNAME allows you to convert the SUFFIX value to hexadecimal. HEXNAME is valid only when PRFLINE, PRFPU, and PRFLU are coded. The default value is YES when PRFLINE, PRFPU, and PRFLU are coded.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names, and how to specify the related statement operands.

```
[,HPRQLIM={nnn}]  
      {0 }
```

X.25 ODLC and NCP V7R5 and later: Specifies the maximum number of APPN HPR data bytes that can be queued to a peripheral link station’s transmission queue at one time. If an APPN HPR frame arrives at a link station which has a transmission queue that is already at or above this limit, the frame is discarded.

The default is 0, which means that there is no limit to the amount of APPN HPR data that can be queued. The maximum is 1677215.

Notes:

1. This keyword is valid for ODLC X.25 stations only.
2. This keyword is supported only when PUTYPE=2 and XID=YES.
3. If HPRQLIM is not coded on the X25.PU or X25.VC statement, and it is coded on the BUILD statement, then the value on the BUILD statement is used.

[,ID={bbbfffff}]

Specifies the ID parameter used to support the NCP anonymous caller function where *bbb* is the IDBLK value and *ffff* is the first IDNUM value on the X.25 VC GROUP. X.25 generates values for each subsequent SVC in increments of 2. If ID is not specified, this value is taken from the IDBLK/IDNUM of the remote SNA DTE.

Note: The ID keyword can only be applied to LLC type 2, LLC type 3 non-SVCSC, and LLC type 0 without subaddressing or DATE. You can either code the ID keyword for all SVCs of an MCH working in LLC type 2, LLC type 3 non-SVCSC, or LLC type 0 without subaddressing or DATE, or do not code ID for any of them.

[,IDNUMT={ffff}]

Specifies a user-defined IDNUM value for the first of a X25.VC group of non-SNA SVCs. Specifies four hexadecimal digits to add to the hexadecimal digit specified by the NCP BUILD X25.IDNUMH keyword to form the IDNUM that NPSI can use for connections with non-SNA SVCs.

For each subsequent SVC defined by the LCN keyword, NPSI generates IDNUM values in increments of 2.

If you do not specify IDNUMT for any SVC, the default is a number that is generated internally by NPSI based on the reverse order of the SVCs in the generation.

The IDNUMT keyword must be an even number.

Notes:

1. You cannot code IDNUMT on an X25.VC statement belonging to a fast connect or FTPI MCH.
2. You must either code the IDNUMT keyword on all non-SNA SVCs for all the MCHs of the entire NPSI generation, or not code the IDNUMT keyword on any of them.
3. These IDNUM values cannot be directly associated to any specific logical channel number (LCN).

[,ISTATUS={ACTIVE }]
{INACTIVE}]

Specifies whether the lines, physical units, and logical units are activated following a VTAM first start or cold restart. There is no default. This keyword and parameter are reported on the generated NCP LINE statements.

[,LLC={LLC0}]
{LLC4}
{LLC5}]

Specifies the PVC type defined by this statement. This LLC keyword is mandatory for PVCs and should be coded as follows:

- Code LLC0 if the virtual circuit connects to a non-SNA DTE and does not require GATE or PAD support (type 0 virtual circuit).

- Code LLC4 if the virtual circuit requires GATE support (type 4 virtual circuit).
- Code LLC5 if the virtual circuit connects to a terminal that requires PAD support (type 5 virtual circuit). The type of PAD support (integrated or transparent) is specified on the PAD keyword of the X25.MCH statement.

[,LLERP=REQUIRED]

X.25 ODLC and NCP V7R5 and later: Specifies whether the link station will perform link level error recovery procedures when sending HPR data on this link. LLERP=REQUIRED is the NCP default and is required for running HPR over X.25 ODLC stations.

[,MAXDATA={number}]
{265 }

Specifies the maximum number of bytes in an SNA segment. The default value is 265. See the *NCP, SSP, and EP Resource Definition Reference* for calculation and more information.

Usage Notes: Because SNA segmenting is not supported for non-SNA LLCs, you need to specify a MAXDATA value greater than the maximum PIU size.

[,MAXOUT=number]

When the value of MAXOUT is not equal to 6, the first data packet received is sent to the SSCP as a logon message. Specify MAXOUT=6 to queue data arriving before start data traffic (SDT) in NPSI for LLC type 0 and LLC type 5. At SDT, this data is passed to the application.

Usage Notes: MAXOUT is coded only for PVCs. For SVCs, use the MAXOUT keyword in the VTAM switched major node.

[,NCPGRP={label}]
{NEW }

Specifies the symbol of the NCP GROUP statement that defines the virtual circuits. If you do not specify this value, the name is generated by NDF. Code this keyword to modify the group name to which the SNA resources generated by the X25.VC belong.

If you specify *label*, a new NCP GROUP is created using this label as the GROUP statement name.

If you specify NEW, a new NCP group is automatically created at NPSI generation.

V3R3 Only: For V3R3, NPSI creates, by default, one group of PVCs, one group of SVCs, and one group of SVCSCs for each X25.MCH statement.

X.25 ODLC and NPSI V3R4 and Later: Four groups are created by default for each X25.MCH definition. One group is created for INN PVCs, another for BNN PVCs, a third for INN SVCs, and a fourth for BNN SVCs.

If you omit the NCPGRP keyword, the virtual circuits belong to the most recently created group name. The default name is generated as follows:

p25Pxxxy for PVCs

p25Sxxxy for SVCs

See “Resource Naming Conventions” on page 5-2 for information on default naming conventions.

Note: This keyword can no longer be used to make a call-out on a virtual circuit belonging to a given set of virtual circuits within an MCH. When the virtual circuits are within an MCH, X.25 allocates the virtual circuits for call-out starting from the largest virtual circuit number that is available for call-out. There is no need to have more than one group associated with BNN PVCs,

one group associated with INN PVCs, one group associated with INN PVCs, one group associated with BNN SVCs, and one group associated with INN SVCs for each X25.MCH definition.

```
[,NGFINC={text string    }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFINC keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.VC statement.

If **ALL**,*text string* is coded, then the NGFINC keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.

If the NGFINC keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFINC on the PU and LU statements generated by the X25.VC statement, code

```
NGFINC=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*      NGFINC=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*      NGFINC=YYYY
```

will be generated following the LU statement.

```
[,NGFOPT={text string    }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFOPT keyword and its text string are copied to the PU statement generated by the X25.VC statement.

If **ALL**,*text string* is coded, then the NGFOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.

If the NGFOPT keyword is desired on specific statements then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFOPT on the PU and LU statements generated by the X25.VC statement, code

```
NGFOPT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*          NGFOPT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*          NGFOPT=YYYY
```

will be generated following the LU statement.

```
[,NGFTXT={text string    }]
      {ALL,text string}
      {GROUP,text string,...LU,text string}
```

This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.

If just the *text string* is coded, then the NGFTXT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.VC statement.

If **ALL**,*text string* is coded, then the NGFTXT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.

If the NGFTXT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the “targ_name,value” group. For example, to code NGFTXT on the PU and LU statements generated by the X25.VC statement, code

```
NGFTXT=(PU,XXXX,LU,YYYY)
```

In the generated NCP statements, a comment keyword of

```
*          NGFTXT=XXXX
```

will be generated following the PU statement, and a comment keyword of

```
*          NGFTXT=YYYY
```

will be generated following the LU statement.

```
[,OSITYPE={number}]
      {0    }
```

Specifies in decimal the GATE CTCP with which these virtual circuits must communicate. This parameter applies to PVCs only, and must be used when the MCH is capable of communicating with several GATE CTCPs. If the MCH is not capable of communicating with more than one GATE CTCP, this keyword must be omitted. Valid values range from 0–27. The default value is 0.

Notes:

1. A PVC can be bound to any CTCP, regardless of the CTCP number and the OSITYPE. If a control packet (for example, RESET) is received from the network and sent over the control session, NPSI uses the OSITYPE keyword to direct the packet to an appropriate CTCP.
2. A GATE MCH can communicate with more than one CTCP when it has SVCs that are controlled by several CTCPs (see the CTCP, CUD0, and SUBD keywords on the X25.MCH statement). However, when an MCH has only PVCs, or PVCs and SVCs controlled by the same CTCP, there can only be one CTCP. When this situation occurs, the OSITYPE keyword must be either omitted, or coded as OSITYPE=0 on the X25.LINE statement.

[,OUFINDX=*index*]

Specifies the default index value in the X25.OUFT table. The index value points to the table entry that builds the facility field and the end of the CUD field in a Call Request packet. OUFINDX serves as a default for call-outs, if you code yy=00 in the DIALNO or DLCADDR keyword. Valid values range from 1–99. This keyword is required when you code TYPE=SWITCHED or TYPE=S. This keyword is not required when you code DSTNODE=INN.

Note: OUFINDX must not be coded for virtual circuits belonging to an FTPI MCH.

[,PRFLINE=*x*]

Specifies a user-defined prefix for the name of the NCP LINE statements generated for this X25.VC statement. This keyword can be a character string of 1–7 characters.

Note: The first character in the PRFLINE keyword must be alphabetic.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names, and how to specify the related statement keywords.

[,PRFLU=*z*]

Specifies a user-defined prefix for the name of the NCP LU statements generated for this X25.VC statement. This keyword can be a character string of 1–7 characters.

Note: The first character in the PRFLU keyword must be alphabetic.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names, and how to specify the related statement keywords.

[,PRFPU=*y*]

Specifies a user-defined prefix for the name of the NCP PU statements generated for this X25.VC statement. This keyword can be a character string of 1–7 characters.

Note: The first character in the PRFPU keyword must be alphabetic.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names, and how to specify the related statement keywords.

[,RESETUNB={*YES*}
NO }

Starting with V3R9: Specifies whether NPSI should send a RESET packet when a BIND or UNBIND is received from the application, or a data packet is received but the PCNE LU session is not bound. NO is the default.

When RESETUNB=YES is coded, MAXOUT=6 must be coded on the X25.VC or X25.PU statement that defines the PCNE PVC.

RESETUNB=YES is allowed only for LLC0 PVCs. RESETUNB=YES is not allowed when GATE=DEDICAT is specified on the X25.MCH statement.

[,RETVCCCT={*count*}
3 }

Specifies the number of retransmissions of the physical services command on virtual circuits connected to an SNA subarea or peripheral node. Valid values range from 0–255; the default value is 3.

[,RETVCTO={*timeout*}
30 }

Specifies the timer used between retransmissions of a physical services command on virtual circuits connected to an SNA subarea or peripheral node. Valid values range from 0–255 seconds; the default value is 30 seconds.

- [,SPAN=*name*]
- Specifies that the NetView program operator's access to the SNA resources generated by the X25.VC statement is restricted. This specification is copied to the LINE, PU, and LU statements generated by the X25.VC statement.
- [,STATOPT=*text string*] **V3R4 - V3R8:** Specifies whether the NetView status monitor is used for this resource. The STATOPT keyword and its text string operand are copied as specified into a comment statement following the LINE statement generated by NDF.
- [,STATOPT={*text string* }]
 {**ALL**,*text string*}
 {**GROUP**,*text string*,...**LU**,*text string*}
- Starting with V3R9:** This keyword will be generated as a comment following the specified NCP statement(s), to be processed by NetView.
- If just the *text string* is coded, then the STATOPT keyword and its text string are copied to the LINE, PU, and LU statements generated by the X25.VC statement.
- If **ALL**,*text string* is coded, then the STATOPT keyword and its text string are copied to the GROUP, LINE, PU, and LU statements generated by the X25.VC statement.
- If the STATOPT keyword is desired on specific statements, then that can be achieved by coding the desired statement name preceding the text string. Up to four targets may be specified by repeating the "targ_name,value" group. For example, to code STATOPT on the PU and LU statements generated by the X25.VC statement, code
- ```
STATOPT=(PU,XXXX,LU,YYYY)
```
- In the generated NCP statements, a comment keyword of
- ```
* STATOPT=XXXX
```
- will be generated following the PU statement, and a comment keyword of
- ```
* STATOPT=YYYY
```
- will be generated following the LU statement.
- [,SUFFIX=*w*]
- Specifies a number that is the lower limit for each set of names built. There is no default. This parameter must be specified together with PRFLINE, PRFPU, and PRFLU if user-defined names are generated. The length of the name is equal to the length of PRFLINE, or PRFPU, or PRFLU, added to the length of the SUFFIX parameter. Values range from 0–X'FFF' when HEXNAME=YES, and from 0–4095 when HEXNAME=NO.
- PRFLINE, PRFPU, PRFLU, and SUFFIX must be coded together.
- Note:** The number of virtual circuits generated is dependent on the number of digits specified in the SUFFIX parameter. For example:
- If you code SUFFIX=1 and HEXNAME=NO, only virtual circuits 1–9 are generated.
  - If you code SUFFIX=1 and HEXNAME=YES, only virtual circuits 1–F are generated.
  - If you code SUFFIX=01 and HEXNAME=NO, only virtual circuits 1–99 are generated.

- If you code SUFFIX=01 and HEXNAME=YES, only virtual circuits 1–FF are generated.

See “Resource Naming Conventions” on page 5-2 for a description of the options available for the automatic creation of resource names, and how to specify the related statement keywords.

[,VTMKYWRD=(*targ\_name*,*kywrд\_name*,*kywrд\_value*)]

Specifies any keyword that is not known to NPSI but needs to be generated for an NCP statement.

*targ\_name* is the name of the statement the keyword is to be generated for. This can be for GROUP, LINE, PU, LU, or ALL.

*kywrд\_name* is the name of the keyword to be generated.

*kywrд\_value* is the value to be coded as an operand for the keyword.



## X25.VCCPT Statement

**Usage:** Use this statement to describe the connection parameters for one or more virtual circuits. At least one X25.VCCPT statement is required; however, up to 99 statements can be coded. One statement is required for each entry on the virtual circuit connection parameter table. Entries need not be consecutive. Table length is specified by the CPHINDX keyword

**Position:** This statement is positioned after the X25.NET statement and before the X25.MCH statement. Each virtual circuit must use an entry in the virtual circuit connection parameter table.

**Format and Keywords:** The format of the X25.VCCPT statement is:  
 symbol      X25.VCCPT      keywords

For virtual circuits used exclusively by LLC type 4 or those under the control of a DATE CTCP, the MAXPKTL and VWINDOW keywords of the X25.VCCPT statements are not used, because these parameters are defined by the CTCP. However, the INSLOW keyword is used in the X25.VCCPT statement referenced by associated virtual circuits.

Table 5-21 shows the keywords that you can code on the X25.VCCPT statement.

Table 5-21. X.25 Keywords for X25.VCCPT Statement

### Keywords

```
INDEX=index,
MAXPKTL=size
[,INSLOW={ (percent1,percent2)}]
 { (50,12) }
[,VWINDOW={value}]
 {2 }
```

**INDEX**=*index*

Specifies the entry of the VCCPT that is initialized by the X25.VCCPT statement. For example, INDEX=3 initializes the third entry in the VCCPT. Valid values range from 1–99.

**MAXPKTL**=*size*

Specifies (in bytes) the maximum length of the packets to be sent or received over the virtual circuits. This length is the maximum length of data in a packet, excluding the length of the packet header. Valid values range from 32–4096 (according to the network subscription). There is no default value. For a subarea node virtual circuit, the MAXPKTL value must be at least 128.

For incoming packets, make sure that the accumulated PIU length does not exceed the maximum data length defined in the NCP HOST statement. See the *NCP, SSP, and EP Resource Definition Reference* for more information on the HOST statement.

For outgoing packets, X.25 splits the PIU data into packets using the M-bit, except for type 2 virtual circuits. For type 2 virtual circuits, NPSI splits the PIU data using segment indication. The maximum packet length equals the packet length defined for the remote DTE.

**MAXPKTL**=(*sizein,sizeout*)

**X.25 ODLC:** This format allows incoming and outgoing packets to have different maximum lengths. For PVCs, *sizein* is the maximum packet size for incoming packets and *sizeout* is the maximum packet size for outgoing packets. For SVCs, *sizein* is the maximum packet size for packets flowing

from the called to calling DTE and *sizeout* is the maximum packet size for packets flowing from the calling to called DTE. If *sizeout* is not specified, it defaults to *sizein*. There is no default for *sizein*.

```
[,INSLOW={(percent1,percent2)}]
 { 50,12 }
```

Specifies the percentage of free buffers that can be reached before the virtual circuits using this number enter an unsafe or danger situation. Unsafe and danger thresholds are mechanisms used by NPSI to avoid NCP slowdown. *percent1* and *percent2* values are specified in relation to the values in the SLOWDOWN keyword of the NCP BUILD statement.

*percent1*      Defines the UNSAFE threshold.  
*percent2*      Defines the DANGER threshold.

Valid values are 0, 6, 12, 25, 50, and 100. When this keyword is coded, both values must be specified, and *percent1* must be greater than *percent2*. Default values of 50 for *percent1*, and 12 for *percent2*, mean that the UNSAFE limit is 50 percent higher while the DANGER limit is 12 percent higher than the NCP SLOWDOWN value.

The INSLOW keyword is ignored for X.25 ODLC.

**Note:** This keyword is not taken into account for subarea node virtual circuits. See the *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for more information.

```
[,VWINDOW={value}]
 { 2 }
```

Specifies the value of the transmit or receive window used by the packet protocol for the virtual circuits that use the entry. This value must be less than the packet modulo value defined in the X25.MCH statement. Valid values range from 1–127; the default is 2.

**For SVC Only:** GATE and DATE do not use the VCCPT to define the packet size and the packet window. Packet size and packet window are defined by the CTCP. Because other keywords are used, at least one X25.VCCPT statement must be coded for compatibility.

**Note:** The value coded on VWINDOW must correspond to the packet window used by the partner. For example, if NPSI acts as a DTE, you must code VWINDOW with a value equal to the packet window used in the DCE.

```
[,VWINDOW={(valuein,valueout)}]
```

**X.25 ODLC:** This format allows incoming and outgoing packets to have different window sizes. For PVCs, *valuein* is the window size for incoming packets and *valueout* is the window size for outgoing packets. For SVCs, *valuein* is the window size for packets flowing from the called to calling DTE and *valueout* is the window size for packets flowing from the calling to called DTE. If *valueout* is not specified, it defaults to *valuein*. The default for *valuein* is 2.

---

## Chapter 6. Coding NCP and VTAM Parameters for X.25 Support

|                                                             |      |
|-------------------------------------------------------------|------|
| NCP Considerations                                          | 6-2  |
| NCP Statements Changed for X.25                             | 6-2  |
| OPTIONS Definition Statement                                | 6-2  |
| LUDRPOOL Definition Statement                               | 6-2  |
| NCP Statements Created by X.25 Generation                   | 6-3  |
| PU Definition Statement                                     | 6-3  |
| LU Definition Statement                                     | 6-4  |
| VTAM Considerations for Switched Major Node                 | 6-6  |
| VTAM User-Exit Routine (X.25 ODLIC and NPSI V3R5 and Later) | 6-6  |
| PU Definition Statement                                     | 6-6  |
| PATH Definition Statement                                   | 6-8  |
| DIALNO                                                      | 6-8  |
| DLCADDR                                                     | 6-10 |
| Using the DIALNO and DLCADDR Keywords                       | 6-12 |
| LU Definition Statement                                     | 6-14 |
| SNA Type 2.1 Node                                           | 6-14 |
| SVCSC Considerations                                        | 6-14 |
| PU Statement for SVCSC                                      | 6-14 |
| PATH Statement Definition for SVCSC                         | 6-15 |
| Call-Out                                                    | 6-15 |
| Call-In                                                     | 6-16 |
| Call-Inout                                                  | 6-17 |
| Correlation of PATH Statements                              | 6-17 |
| Session Continuation                                        | 6-18 |
| Virtual Route Pacing for Subarea-to-Subarea Communication   | 6-19 |

This chapter explains the following considerations that are specific to X.25:

- NCP
- VTAM
- SVCSC
- Session continuation
- Virtual route pacing for subarea-to-subarea communication

### NCP Considerations

When you code NCP statements to work with X.25, consider the following:

- NCP statements changed for X.25
- NCP statements created by X.25 generation

### NCP Statements Changed for X.25

This section provides information about NCP statements that must be coded differently for X.25.

#### OPTIONS Definition Statement

You must code the OPTIONS definition statement as the first statement in your NCP generation statement list. The two required keywords are USERGEN and NEWDEFN.

The USERGEN keyword indicates the name of the NDF load module that contains X.25 generation modules. Code it as:

```
USERGEN=X25NPSI for NCP V7R3 and earlier
USERGEN=CBEX25 for NCP V7R4
```

If you omit USERGEN, X.25 statements are not processed. Any X.25 statements found during the NCP generation are flagged as errors.

You must code NEWDEFN=YES on the OPTIONS statement so that NDF creates a new generation definition for input to VTAM. This new definition consists of the NCP generation definition and keywords passed from the generation load modules specified on USERGEN.

Externally, the X.25 statements appear in the definitions passed to VTAM; however, VTAM ignores these statements because of the X25. prefix. VTAM can process these statements after they are generated into a new definition file containing definitions created from the statements with the X25. prefix.

You should code NEWDEFN=(YES,ECHO,SUPP). The ECHO value causes the statements in the new definition file to be in a format that is similar to the original file. The SUPP value eliminates the comment, GENERATED BY X25, that prefixes the statements generated by NDF in the new definition file.

#### LUDRPOOL Definition Statement

The LUDRPOOL definition statement must be specified in the NCP when switched virtual circuits are used. The statement keywords NUMTYP1, NUMTYP2, and NUMILU must be coded. These keywords specify the number of sets of logical unit control blocks to be available for dynamic allocation to switched connections.

NUMTYP1 and NUMTYP2 specify the sets of logical control blocks required when the logical unit is associated with a PU type 1 and a PU type 2, respectively. NUMILU specifies the sets of logical unit control blocks for independent LUs for sessions involving SNA type 2.1. The sessions supported in the value coded for NUMILU can exist between SNA type 2.1 nodes and other SNA type 2.1 nodes.

## NCP Statements Created by X.25 Generation

The NCP GROUP, LINE, PU, and LU definition statements are generated for the X25.MCH, X25.VC, and the X25.FCG statements. Examples are provided to help you decide which types of circuits best fit your site's requirements.

**X.25 ODLC and NPSI V3R5 and Later:** Code PHYSRSC=YES to make NCP and VTAM aware of the association between this MCH and its associated virtual circuits. This ensures that NCP and VTAM process the MCH and its associated virtual circuits as one complex link (delaying virtual circuit link activation until the MCH link and the PU are active, rejecting MCH PU deactivation if the virtual circuits are still active, and displaying the MCH name when a virtual circuit is displayed). If PHYSRSC is not coded, there is no default value, and the MCH and associated virtual circuits are still processed as independent resources, not as one complex link.

### PU Definition Statement

The following examples show the PU definitions normally generated by X.25 for physical circuits, PVCs, and SVCs, respectively:

- Table 6-1 illustrates the PU statement generated for a physical circuit.
- Table 6-2 on page 6-4 illustrates the PU statement generated for a PVC.
- Table 6-3 on page 6-4 illustrates the PU statement generated for an SVC.

*Table 6-1. PU Statement for a Physical Circuit*

| Name        | Operation | Keywords                                                                                                                                     |
|-------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------|
| name        | PU        | ADDR=01,<br>ANS=...,<br>MAXDATA=265,<br>PUTYPE=1                                                                                             |
| name        |           | X.25 assigns a physical unit name during the generation process or takes one from the PUNAME keyword of the X25.MCH statement.               |
| ADDR=01     |           | Specifies the hexadecimal representation of the 8-bit address of the physical unit represented by this PU statement. The address code is 01. |
| ANS=...,    |           | Copied from the X25.MCH statement.                                                                                                           |
| MAXDATA=265 |           | Specifies the maximum length of a PIU sent to a type 1 PU.                                                                                   |
| PUTYPE=1    |           | Specifies the type of physical unit of the SDLC station represented by this PU statement instruction. The PU type is 1.                      |

When a PVC is defined by an X25.VC statement, the PU generated is for type 0, 4, or 5 virtual circuits.

Table 6-2. PU Statement for a Permanent Virtual Circuit

| Name | Operation | Keywords                          |
|------|-----------|-----------------------------------|
| name | PU        | ADDR=01,<br>PACING=1,<br>PUTYPE=1 |

You can code MAXDATA and VPACING in the X25.VC statement in your X.25 generation statements.

For type 2 or 3 virtual circuits use the X25.PU statement.

When an SVC is defined by an X25.VC statement, the PU generated will be:

Table 6-3. PU Statement for a Switched Virtual Circuit

| Name | Operation | Keywords     |
|------|-----------|--------------|
| name | PU        | PUTYPE=(1,2) |

X.25 does not support the SRT keyword of the PU statement. Default values for the SRT thresholds are 32768 for counters contained in 2 bytes, and 128 for counters contained in 1 byte. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for the format of these counters within RECFMS.

### LU Definition Statement

Several LU statements are generated for the physical circuit if you code GATE=GENERAL and if virtual circuits of the MCH need to communicate with several CTCPs. You can specify up to 28 CTCPs.

The following examples show the LU definitions normally generated by NPSI

- Table 6-4 illustrates the first LU statement generated for a physical circuit with GATE=NO.
- Table 6-5 on page 6-5 illustrates how subsequent LU statements are generated for a physical circuit.

Table 6-4. First LU Statement for a Physical Circuit

| Name | Operation | Keywords                       |
|------|-----------|--------------------------------|
| name | LU        | LOCADDR=0,<br>ISTATUS=INACTIVE |

LOCADDR=0 Specifies the local address of the logical unit in decimal notation without leading zeros. Code it as 0.

ISTATUS=INACTIVE Informs VTAM that this logical unit should not be activated after initial program load (IPL) if GATE=NO is coded in the X25.MCH statement.

Table 6-5. Subsequent LU Statement for a Physical Circuit

| Name   | Operation | Keywords                     |
|--------|-----------|------------------------------|
| symbol | LU        | LOCADDR=n,<br>ISTATUS=ACTIVE |

For type 0, 4, or 5 virtual circuits, NDF generates one LU statement for each virtual circuit defined with an X25.VC statement that is coded TYPE=PERMANENT. Table 6-6 illustrates the LU generated for a PVC.

Table 6-6. LU Statement for a Permanent Virtual Circuit

| Name   | Operation | Keywords  |
|--------|-----------|-----------|
| symbol | LU        | LOCADDR=0 |

For type 2 and 3 virtual circuits that are connected to SNA peripheral nodes, use the X25.LU statement. However, you cannot specify the X25.LU statement for virtual circuits connected to the subarea node.

For logical units on SVCs, define the logical unit in a switched major node of the host access method.

**For Start/Stop Terminal Users:** Depending on your application, when a PVC is used to communicate with a start/stop terminal (TWX device), you must consider the following information for GROUP and LU macros.

If the Start/Stop terminal is using a PVC link, you may want to code the following in the NEWDEFN supplied to VTAM in SYS1.VTAMLST:

- SSCPFM=USSNTO
  - Code SSCPFM=USSNTO in the LU macro to specify that the simulated LU uses character-coded RUs (with the carriage return as the delimiter) in the SSCP-LU session.
- TERM=TWX
  - Code TERM=TWX in the LU macro to ensure that a carriage return and a linefeed character are sent at the end of each output RU from applications such as TSO.
- VIRTUAL=YES.
  - Code VIRTUAL=YES in the GROUP macro to define the group of PVCs that are supported by the LLC type 5 (PAD) function. If this is not coded, the TERM=TWX operand is ignored, and you have either a carriage return or linefeed problem, or both, with your output data stream.

You must add VIRTUAL=YES to the NEWDEFN supplied to VTAM in the VTAMLST *after* NPSI and NCP generation is complete. If VIRTUAL=YES is specified in the GROUP macro before NCP system generation, the assembly indicates that you must specify LINEFVT, PUFVT, and LUFVT. This is not valid for the NPSI and NCP environment. Including VIRTUAL=YES after NPSI and NCP generation provides effective use of this operand as a VTAM only macro, and directs VTAM not to reject session requests for this TWX device. If you do not specify VIRTUAL=YES, the session is rejected by the SSCP when session initiation begins.

---

### VTAM Considerations for Switched Major Node

Use the following guidelines when coding the parameters required for defining an SNA switched major node.

### VTAM User-Exit Routine (X.25 ODLC and NPSI V3R5 and Later)

VTAM V3R4 and later allows you to code a user-exit that is initiated when VTAM receives a REQCONT (request contact) and does not find a SMN PU entry with the same IDNUM and IDBLK values contained in the REQCONT. The user-exit assigns the calling DTE address (which is appended to the REQCONT) to a unique PU, and dynamically creates an entry in the SMN. The SMN PU entry only exists for the duration of the call. See *X.25 NPSI Version 3 Host Programming* for details.

The calling DTE address appended to the REQCONT, in conjunction with the VTAM user-exit routine, now identifies non-SNA terminals. This function does not apply for non-SNA LLCs that use a default or user-assigned IDNUM value to identify the SMN PU entry. For those connections, an additional identification can be set in the Call User data field (CUD) and checked by the exit.

### PU Definition Statement

Code the following parameters on the PU definition statement:

- ADDR

The ADDR keyword is always required. It has significance only for SNA peripheral connections. The ADDR keyword may be required by the peripheral node. For example, the 3174 does not use the Address field in qualified packets. It is not used for either SNA subarea connections or for non-SNA connections where NPSI simulates the presence of the link station represented by the PU statement. If the ADDR keyword is not used, any valid value can be specified. However, it is recommended that you specify ADDR=01. It is coded as a 2-digit hexadecimal address with a valid value of 01 or a range from X'01'—X'FE'.

- DISCNT

Code DISCNT=YES if the SVC is to be cleared when the last LU-LU session, which involves the PU, terminates.

DISCNT=YES must be coded for virtual circuits using GATE or DATE, or Integrated PAD using the INVITATION TO CLEAR option (SHUTD=INVCLR coded on the X25.MCH statement).



- **MAXDATA**

For SNA connections, code the MAXDATA keyword according to the maximum PIU segment size, as defined by the documentation that describes the controller.

For non-SNA connections, code the MAXDATA keyword with a value that is equal to or greater than the maximum PIU size that is to be sent.

- **MAXOUT**

When the value of MAXOUT is not equal to 6, the first data packet received is sent to the SSCP as a logon message. Code MAXOUT=6 when using LLC type 0 and LLC type 5 to cause NPSI to queue all data received before SDT. At SDT, this data is passed to the application.

- **PUTYPE**

Code the PUTYPE keyword as required by the remote PU when connected using LLC type 2 and 3.

Code PUTYPE=1 for LLC types 0, 4, or 5.

- **IDBLK**

Code the IDBLK keyword as required by the remote PU when using LLC types 2 or 3.

See Chapter 4, “Defining X.25 Networks” for more programming details regarding the IDBLK keyword.

- **IDNUM**

Code the IDNUM keyword as required by the remote PU when using LLC type 2 or 3.

See Chapter 4, “Defining X.25 Networks,” for more programming details regarding the IDNUM keyword.

- **CPNAME (SNA type 2.1)**

CPNAME specifies the control point (CP) name of an SNA type 2.1 peripheral node. CPNAME is valid only for type 2.1 nodes that have a CP name. In a switched environment, VTAM can locate the appropriate PU/LU pair by using either the normal IDNUM/IDBLK combination or through the use of the CPNAME keyword on the PU statement in a switched major node source statement member of VTAM.

### PATH Definition Statement

The VTAM PATH statement defines a *path* for an incoming or outgoing call. The PATH statement is optional for incoming calls, but is required for outgoing calls.

For incoming calls, the PATH statement can be used to ensure that the call originates from an authorized destination. For outgoing calls, the PATH statement supplies the information required by VTAM and NCP to initiate the call. X.25 uses the DIALNO or DLCADDR keywords of the PATH statement for outgoing calls only.

If you do not need the expanded capabilities of the DLCADDR keyword, you can code the dial digits information on the PATH statement as shown under “DIALNO.”

If you need the expanded capabilities of the DLCADDR keyword, enter the dial digits information on the PATH statement as shown under “DLCADDR” on page 6-10.

You can code both DIALNO and DLCADDR on the PATH statement. If you code both DIALNO and DLCADDR on the PATH statement and the NCP is a release prior to V7R1 or later, VTAM uses the DIALNO information and works in migration mode. If the NCP is at V7R1 or later, VTAM uses the DLCADDR information.

#### DIALNO

DIALNO is used to establish an outgoing call connection for virtual circuit types 0, 2, 3, or 5 without DATE. DIALNO must be coded in one of the following formats:

```
DIALNO=NNN...N[*MMM...M]LXXYY[*ZZZZZ]
```

```
DIALNO=NNN...N[*MMM...M]LXXYY*DDD[WWWWW] (3710 only)
```

The maximum length of the DIALNO parameter is 32 characters, including asterisks. If you need more than 32 characters, see “DLCADDR” on page 6-10.

- NNN...N

Is the network address of the called DTE. Up to 15 characters can be used.

- MMM...M

Is the network address of the calling DTE. Up to 15 characters can be used. This is an optional field; however, it is mandatory when the PSDN requires that the calling DTE address be specified.

This field is also required when SVCSC is used and the packet switching network does not add the calling DTE address to the call packet.

- L

Specifies a one-digit virtual circuit type code that defines the LLC type (qualified by the node type being accessed for LLC type 3), and the CUD field for LLC types 0 and 5 used by the connection being established. See Table 4-9 on page 4-27 for more information.

- XX

Is the 2-digit VCCPT index that has valid values ranging from 00—99.

XX=00 commands X.25 to use the values defined in the VCCPT entry that are coded in the VCCINDX keyword of the X25.VC or X25.LINE statement.

XX=xx commands the packet procedure to use the values defined in the VCCPT entry with INDEX=xx. You must define this entry during X.25 generation.

- YY

Is the 2-digit OUFT index that can have values ranging from 00—99.

YY=00 commands X.25 to build the Call Request packet with the facility field and the user data that is specified using the OUFINDX keyword of the X25.VC or X25.LINE statement.

YY=yy commands X.25 to build the Call Request packet with the facility field and the user data that is defined using the OUFT entry with INDEX=yy. You must define this entry during X.25 generation.

- ZZZZZ

Is the 5-character identification number specified in the IDNUM keyword for the PU statement. This field must be preceded by an asterisk and is valid and required only when you specify a type 0 or 5 virtual circuit.

If you use this field to specify a default IDNUM, enter the following special characters for the hexadecimal digits A—F.

|   |       |
|---|-------|
| : | for A |
| . | for B |
| % | for C |
| _ | for D |
| ; | for E |
| ? | for F |

- DDD

Is the 3-digit decimal upstream address (USA) of the terminal addressed by the outgoing call. Valid DDD values are 000—255.

This field is used only by the IBM 3710 network controller and applies only to virtual circuit type 3 (L=3). The IBM 3710 network controller, which has an internal table that converts the USA into the actual terminal address, uses this field to determine the specific terminal that the call addresses.

- WWWW

Is the 5-digit decimal identifier number of the physical unit used to map the terminal addressed by the outgoing call. This field only applies to the IBM 3710 network controller and is used to return the IDNUM to the host when the host requests the PU XID. The IDNUM keyword of the PU statement in the SMN must equal this value. The identifier block is 044 and is automatically generated by the IBM 3710 network controller.

**Note:** WWWW does not apply to the IBM 3710 when it is attached to a real type 2 physical unit. In this instance, it should be omitted because the physical unit provides the type 2 XID. The WWWW digits are replaced by five 0 digits in the CUD part of the Call packet. The CUD length taken by the 3710 is constant.

A coding example of the DIALNO parameter for a virtual circuit using LLC type 3 for an IBM 3710 is:

```
DIALNO=17500023530202*00425184
```

## Coding NCP and VTAM Parameters

In this example, the network address of the called DTE is 175000235, the VCCPT and OUTFT entries are 2, and the called cluster is an IBM 3710 network controller with an upstream address of 004 and an IDNUM of 25184. If the OUFT entry for INDEX=2 has OPTFACL=420707 and USRFILD=1234, and the virtual circuit number is 005, the corresponding Call Request packet is built as follows:

```
X'10050B09175000235003420707C30100008004251840000001234'
```

where:

|              |                                                                  |
|--------------|------------------------------------------------------------------|
| 1            | Is the general format identifier.                                |
| 0            | Is the logical channel group number.                             |
| 05           | Is the logical channel number.                                   |
| 0B           | Is the packet type identifier (Call Request).                    |
| 0            | Is the length of calling DTE address.                            |
| 9            | Is the length of called DTE address.                             |
| 175000235    | Is the called DTE address.                                       |
| 0            | Is the padding for even number of bytes.                         |
| 03           | Is the facility length.                                          |
| 420707       | Is the facility.                                                 |
| C3010000     | Is the virtual circuit type 3.                                   |
| 8            | Is the length of USA and IDNUM.                                  |
| 004          | Is the USA.                                                      |
| 251840000000 | Is the IDNUM to be returned by the 3710, plus 7 reserved digits. |
| 1234         | Is the end of CUD.                                               |

The called and calling DTE addresses are concatenated by filling 1 digit for each halfbyte. If the total number of digits is odd, then the concatenated addresses are padded on the right with a 0. Padding is not done between the called DTE address and the calling DTE address.

**X.25 ODLC and NPSI V3R5 and Later:** The calling DTE address is passed to VTAM in a control vector (CV25) appended to the REQCONT. If you code VERIFY=IN on the PATH statement, you must code VERID with the authorized calling DTE address or list of addresses. When VTAM locates the SMN PU that matches the PU entry coded on the IDNUM and IDBLK, it compares the calling DTE address to the address or list of addresses coded on VERID. If no matching address is found, the connection is abandoned.

**Note:** When a default or a user-assigned IDNUM is coded (PAD, GATE, PCNE under DATE, or PCNE under GATE with SUBADDRESSING), this function does not apply.

### DLCADDR

For X.25 connectivity, additional addressing information is sometimes required in addition to the existing addressing information specified using the DIALNO operand on the VTAM PATH statement. VTAM allows you to enter the dial information through the DLCADDR keyword on the PATH statement in the VTAM switched major node. The VTAM DIALNO keyword is limited to 32 characters. The DLCADDR keyword eliminates this 32-character restriction.

The DLCADDR keyword allows information formerly stored in the NCP/NPSI generation to be coded on the PATH statement. This helps reduce communication controller down time that was needed to update NCP generations.

You must determine the DLCADDR subfield number authorized for X.25, the information contained in the subfield, and the format in which the information must be entered.

Code the DLCADDR keyword as follows:

**DLCADDR**=(*subfield\_id,data\_type,data\_string*)

where:

**subfield\_id**

Specifies the subfield identification number. Subfields 1 and 20 are mandatory. Subfield\_id values are shown in Table 6-7.

Table 6-7. DLCADDR Subfield ID Values

| Subfield | Subvector   | Data String                                                        |
|----------|-------------|--------------------------------------------------------------------|
| 1        | X'91'       | DLC type identifier (C'X25' for NPSI)<br>(C'X25SVC' for X.25 ODLC) |
| 20       | X'A4'       | Has the same format used in the DIALNO keyword                     |
| 21       | X'A5'       | Called DTE Address (X.25 ODLC Only)                                |
| 22       | X'A6'       | Calling DTE Address (X.25 ODLC Only)                               |
| 30       | X'AE'       | User facilities                                                    |
| 40-49    | X'B8'-X'C1' | Calling DTE address extension                                      |
| 50-59    | X'C2'-X'CB' | Called DTE address extension                                       |
| 60       | X'CC'       | Offset in CUD where data must be placed                            |
| 61-69    | X'CD'-X'D5' | CUD data                                                           |

**data\_type**

You can choose one of the following data types:

- C** Specifies character. You can specify 1–250 characters.
- A** Specifies compressed alphanumeric. You can specify a maximum of 332 characters.
- BCD** Specifies binary coded decimal. You can specify a maximum of 500 digits.
- X** Specifies hexadecimal. You can specify a maximum of 500 digits.
- D** Specifies decimal. You can specify a maximum of 10 digits.

**data\_string**

You can choose one of the data strings shown in Table 6-7.

You must code one DLCADDR keyword for each piece of information required to establish the call. For example, if you use the DLCADDR keyword enter the information on the PATH statement in the VTAM switched major node as follows:

## Coding NCP and VTAM Parameters

```
DLCADDR=(1,C,X25),
DLCADDR=(20,C,nnn.n*mmm.mLXXYY*ZZZZ),
DLCADDR=(30,X,420707430404),
DLCADDR=(40,C,CALLING ADD EXT),
DLCADDR=(41,X,0001),
DLCADDR=(50,C,CALLED ADD EXT),
DLCADDR=(51,X,0001),
DLCADDR=(60,D,1),
DLCADDR=(61,C,CUD COMMENT)
```

**Note:** You can also code the DIALNO parameter.

VTAM analyzes DLCADDR keyword information and passes it to NCP through subvectors imbedded in control vector X'69'. VTAM creates one subvector per subfield and imbeds them in control vector X'69', which is appended to the CONNOUT PIU. Control vector X'69' has a maximum length of X'FF' (255). See Table 6-7 on page 6-11 for correspondence between subfield values and subvector numbers.

There are 252 bytes available for subvectors. If there is only one subvector, it can have up to 250 bytes of data (subtract two for the subvector header). For multiple subvectors on a CV69, each additional subvector reduces the available bytes by two.

### Using the DIALNO and DLCADDR Keywords

If you do not need the expanded DIALNO capabilities provided by the DLCADDR keyword, you can enter the dialing information on the PATH statement as shown in the following samples:

#### **Using DIALNO:**

```
PATH . . . ,DIALNO=nnn.n*mmm.mLXXYY*ZZZZ
```

#### **Using DLCADDR:**

```
PATH . . . ,DLCADDR=(1,C,X25), DLCADDR=(20,C,nnn.n*mmm.mLXXYY*ZZZZ)
```

#### **Using both DIALNO and DLCADDR:**

```
PATH . . . ,DIALNO=nnn.n*mmm.mLXXYY*ZZZZ,
 DLCADDR=(1,C,X25),
 DLCADDR=(20,C,nnn.n*mmm.mLXXYY*ZZZZ)
```

When you code both the DIALNO and DLCADDR keywords, the DIALNO information is used by VTAM for versions of NCP prior to V7R1 and the DLCADDR information is used for NCP V7R1 and later releases.

If you need the expanded DIALNO capabilities, enter the dialing information on the PATH statement as shown in the following samples:

**Using the DLCADDR keyword:**

```
PATH ... ,DLCADDR=(1,C,X25),
DLCADDR=(20,C,nnn.n*mmm.mLXXYY*ZZZZ),
DLCADDR=(30,X,420707430404),
DLCADDR=(40,C,CALLING ADD EXT),
DLCADDR=(41,X,0001),
DLCADDR=(50,C,CALLED ADD EXT),
DLCADDR=(51,X,0001),
DLCADDR=(60,D,1),
DLCADDR=(61,C,CUD COMMENT)
```

**Using both the DLCADDR and DIALNO keywords:**

```
PATH ... ,DIALNO=nnn.n*mmm.mLXXYY*ZZZZ,
DLCADDR=(1,C,X25)
DLCADDR=(20,C,nnn.n*mmm.mLXXYY*ZZZZ),
DLCADDR=(30,X,420707430404),
DLCADDR=(40,C,CALLING ADD EXT),
DLCADDR=(41,X,0001),
DLCADDR=(50,C,CALLED ADD EXT),
DLCADDR=(51,X,0001),
DLCADDR=(60,D,1),
DLCADDR=(61,C,CUD COMMENT)
```

If VTAM encounters a back-level NCP (prior to V7R1), coding both the DIALNO and DLCADDR keywords allows VTAM to issue a CONNOUT PIU in the previous format (without subvectors).

X.25 ODLC supports TOA/NPI addresses as specified by ITU-T 93. TOA/NPI addresses must be coded in DLCADDR subfield 21 and, optionally, in subfield 22. Enter the dialing information on the PATH statement as shown in the following example:

```
Called DTE address: 2112345
Calling DTE address: 2154321
LLC type: BNN QLLC (LLC3)
VCCPT index: 1
OUFT index: 1
```

Note: The first byte of a TOA/NPI address is the type byte.

```
PATH ... ,DLCADDR=(1,C,X25SVC),
DLCADDR=(20,C,10101),
DLCADDR=(21,BCD,800721123450),
DLCADDR=(22,BCD,800721543210)
```

Note: Only LXXYY digits are required when subfield 21 is used.

### LU Definition Statement

For type 0, 4, or 5 virtual circuits, you must code PACING=1 (the default value) and LOCADDR=0. If you are using a TWX terminal, you might, depending on the application, have to code TERM=TWX in the LU statement to obtain the correct formatting.

The default IDNUM scheme is identical for virtual circuits whether they are defined as GATE, PAD, or PCNE using subaddressing; therefore, you cannot tell which SNA virtual circuit type corresponds to a PU or LU definition. Do not specify parameters in your PU and LU definitions that will prevent virtual circuit types using an SMN entry from working. For example, USSTAB is often used for PAD definitions, but USSTAB prevents a GATE virtual circuit from working. If USSTAB is required for PAD support or subaddressed PCNE virtual circuits, do not use GATE on the same physical circuit.

### SNA Type 2.1 Node

An SNA type 2.1 node supports connectivity between independent LUs without the need for SSCP involvement. An LU type 6.2 operating in this type of SNA node supports concurrent parallel sessions, as well as multiple concurrent sessions with other LUs.

In a switched environment, VTAM can locate the appropriate PU definition in an SMN using either the normal IDNUM/IDBLK combination or the CPNAME on the PU statement in a switched major node source statement member of VTAM. When using casual connection, CPNAME is the CONTROL POINT name for a peripheral Type 2.1 node, and is the adjacent VTAM name for a VTAM/NCP/NPSI configuration.

---

### SVCSC Considerations

This section lists the keywords that must be coded on various statements in the SMNs for the SVCSC function.

### PU Statement for SVCSC

The following are the keywords that must be coded on the PU statement in the SMN for the SVCSC function.

**SUBAREA** Specifies the subarea number that must match the subarea number obtained from the XID data returned by the contacted subarea node link station.

**TGN** Specifies the transmission group number (TGN) that must match the TGN obtained from the XID data returned by the contacted subarea node link station when ANY is not specified. If ANY is specified, the TGN used to define the routes serviced by this connection is the number returned from the contacted subarea node link station.

The TGN included in the XID sent to the contacted subarea node link station is the TGN coded in the PU statement used for the outgoing call.

**IDNUM** Specifies the identification number (IDNUM) that must match the IDNUM obtained from the XID data returned by the contacted subarea node link station. The IDNUM included in the XID that is sent to the



contacted subarea node link station is the IDNUM that is coded in the PU statement used for the outgoing call.

The IDNUM acts as a correlation number for two PU X.25 statement definitions in the contacting subarea nodes. There can be only one link with the same TGN between two subarea nodes. Therefore, IDNUM is used for SDLC links, leased or switched, where there can be multiple links in the transmission group between two subarea nodes.

- PUTYPE** Specifies the type of the contacted node. You must specify PUTYPE=4 for a contacted NCP node or PUTYPE=5 for a contacted host node (ES/9370 with a TSC) with X.25 support provided by VTAM.
- NETID** Specifies the network identifier of the network to which the contacted node belongs. This field must also match the NETID obtained from the XID data returned by the contacted subarea node link station.

### Notes:

1. IDBLK is not used and does not need to be specified on the PU statement. It takes the value X'FFF'.
2. If the connection is to remain active when there is a failure of the owning SSCP, ANS=CONT or CONTINUE must be specified. The link station can then be activated by a backup SSCP or the original SSCP, and still remain active. The supporting link and the SMN containing the link station (PU) specifications must be activated first.

## PATH Statement Definition for SVCSC

When using SVCSC, with or without the SHM function, special considerations are required for the PATH statement. The PU statement can be defined as CALL=IN, CALL=OUT, or CALL=INOUT using the CALL keyword to prevent call-ins or call-outs for security reasons.

A key parameter of the PATH statement is specified in the GRPNM keyword. It is used to correlate the SMN entry with the X.25 generation for incoming calls as well as call requests. X.25 generates a GROUP of SVCSC LINE/PUs for each MCH by default. You can choose to create several GROUPs of SVCSC for each MCH by coding the NCPGRP keyword in the X25.LINE statements. This will complicate the coding of the SMN, because several PATH statements will be required for each PU in the cases of incoming and outgoing calls.

### Call-Out

To perform a call-out using SVCSC, consider the following:

- The GRPNM keyword must be coded to locate the correct NCP GROUP of the X25.MCH statement where the call will be made.
- If the SNA resource is restricted to call-out only, the PATH statement should be coded as CALL=OUT.
- If SHM is used for this resource, the SHM and SHMTIM keywords must be coded. SHM is not supported by X.25 ODLC.

## Coding NCP and VTAM Parameters

- If the PATH statement is coded with SHM=YES, the corresponding X25.MCH statement must be coded the same to permit SHM implementation. If the coding does not match, VTAM does not perform the call-out.
- In the DIALNO parameter, L specifies the virtual circuit type used by the current connection and it must be coded as L=1. The XX and YY parameters must be coded to designate the VCCPT and OUFT entries.

The DIALNO keyword is required when SHM=YES is coded on the PATH statement.

To ensure security during SVCSC reconnection process, NPSI limits the use of subaddressing versus Short Hold Mode. When the SVCSC call is sent to a remote DTE using subaddressing (another NPSI MCH with SUBADDR=YES), the following situations can occur:

- If the PATH statement is coded with SHM=NO, an extra digit can be added to the called DTE address coded on the DIALNO keyword.
- If the PATH statement is coded with SHM=YES, a digit cannot be added to the called DTE address. When an Incoming Call is received, the call is routed using the CUD field of the Incoming Call packet.

A called DTE address is required when there is no PSDN, for example, when two NCPs are directly connected to each other, or one multichannel is defined as STATION=DCE, and the other as STATION DTE.

For NPSI SVCSC, DIALNO is required and DLCADDR is not allowed. VTAM checks these conditions at switched major node activation.

### Call-In

To perform a call-in using SVCSC, consider the following:

- The PATH statement, if required, is selected as a function of the GRPNM that specifies the NCP LINE over which the call came in. PATH statements are required if SHM, security checking, or CALL=IN is specified for this PU.
- If the SNA resource is restricted to call-in only, the PATH statement should be coded as CALL=IN.
- GRPNM must be coded if CALL, VERIFY, or SHM are coded.

If the call-in can come in through one or more virtual circuit lines associated with NCP GROUP statements, all the GROUP names must be coded in the PATH statements that follow the called PU statement. One PATH statement must be coded for each GROUP of SVCSC over which an incoming call can be made.

- If the SHM function is to be used, the DIALNO, SHM, and SHMTIM keywords must be coded.

The SHMTIM keyword value does not have to be equal on both sides in the PATH statements. NPSI uses the largest value coded for the disconnection. The SHMTIM keyword value is expressed in seconds, and the range is from 0—5400 seconds. The default value is 30 seconds. The SHM disconnection is initiated from the primary side only (the primary side having the highest subarea number). Reconnection can be triggered from either side.

The GRPNM keyword must also be coded if SHM is used. VTAM verifies that the call came on a GROUP is coded SHM. If SHM=NO is coded, no SHM

occurs during the call. To have SHM, SHM=YES must be coded in the following four statements:

- PATH of the calling out side
- GROUP of the NCP calling out side
- GROUP of the NCP called side
- PATH of the called side

This requirement is valid even when the virtual circuit is incoming to the local DTE to be able to construct the Call Request at SHM reconnection.

- If VTAM is to perform security checking for the SNA resource, the following must be coded:
  - VERIFY=IN must be coded.
  - GRPNM must be coded to check if the correct remote DTE identifier is specified in the PATH statement designated by the GRPNM through which the call-in came.
  - One or more DTE addresses must be placed on the VERID keyword. The VERID keyword is valid on call-in requests only.

### Call-Inout

When a PATH statement supports both call-in and call-out and SVCSC is used, the following must be coded:

- The CALL keyword should be coded as CALL=INOUT. The default is CALL=INOUT.
- The GRPNM keyword must be coded to be able to select an NCP group of lines for making the call-out, and for use in SHM definition and security checking for incoming calls.
- If SHM is used, the DIALNO, SHM, and SHMTIM keywords must be coded.
- If VTAM is to perform security checking for the SNA resource, the following must be coded:
  - VERIFY=IN must be coded.
  - One or more DTE addresses must be placed on the VERID keyword.
  - GRPNM must be coded to check if the correct remote DTE identifier is specified in the PATH statement designated by the GRPNM through which the call-in came.

DLCADDR is not allowed for NPSI SVCSC.

### Correlation of PATH Statements

Several PATH statements can be specified for a given SVCSC PU statement. If several are specified, there must be some consistency between them. A PATH statement must be defined for each NCP GROUP used if one of the PATH statements has the following keywords coded:

- CALL=

If the CALL keyword is coded in any of the PATH statements, the call is accepted or performed only if the specification on the PATH designated by the GRPNM corresponds to the attempted operation.

## Coding NCP and VTAM Parameters

- VERIFY=

If VERIFY is coded in any of the PATH statements, and if VERIFY=IN or VERIFY=INOUT is coded in the PATH designated by the GRPNM, the call is accepted only if the VERID contains the caller's address.

See Appendix B, "System Generation Examples," for coding examples.

See the following chapters in *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning*:

- "Error Determination and Recovery" for more details on the SVCSC function
- "Problem Resolution Examples" if a problem occurs
- "NPSI Flow Sequences" for NPSI data flows
- "X.25 ODLC Flow Sequences" for X.25 ODLC data flows

---

## Session Continuation

Session continuation allows LU-LU sessions to remain active when the owning SSCP is lost. When the resource that served to establish ownership of the NCP fails, the NCP enters ANS. The NCP is then no longer owned. Ownership cannot be established on NCP initiative; therefore, the initiative must come from an eventual owner, for example an SSCP.

By coding ANS=CONT at system generation (for PVCs) or in the SMN (for SVCs), a connection is provided with session continuation.

Line, PU, and LU ownership can be taken over by another SSCP at ANS time, and given back to the original SSCP without disruption of the LU-LU sessions, if the partners remain reachable. This applies to SNA PVCs and SVCs and to non-SNA PVCs and SVCs.

ANS=CONT must be coded on the X25.MCH statement to allow CONT to work on the dependent virtual circuits. When ANS occurs, the NCP does not disconnect the PU coded with ANS=CONT. The NCP allows the LINE and PU to remain active so that the LU-LU sessions belonging to the PU are not disturbed. Thus, when the owning SSCP is lost, but the application and remote DTE are not, the data exchange can continue without any disruption.

---

## Virtual Route Pacing for Subarea-to-Subarea Communication

Virtual route pacing window size (VRPWS) must be considered when subarea node virtual circuits are used. For example, the VRPWS value must be larger than the default values to avoid frequent:

- Isolated virtual route pacing responses. These responses use as much NPSI processing time as a normal data packet. Too many of these can increase central control unit (CCU) utilization.
- Entering and exiting virtual route hold. This activity degrades response time on the virtual route. If the CCU is fully loaded, the execution of some background tasks can be delayed because NPSI subarea node tasks have a higher priority.

The *INSLOW* keywords of the *X25.VCCPT* statement are not taken into account for subarea node virtual circuits. Instead, global flow control between subarea nodes controls buffer levels by way of the virtual route pacing mechanism. Therefore, the VRPWS must be adjusted as a function of both the buffer pool size and the *SLOWDOWN* threshold.

See the *NCP, SSP, and EP Diagnosis Guide* under the heading “Network Flow Control Problems” for assistance in calculating the appropriate virtual route window size.



---

## Chapter 7. Controlling X.25 Resources

|                                                                               |      |
|-------------------------------------------------------------------------------|------|
| SNA Resources . . . . .                                                       | 7-2  |
| Permanent Virtual Circuits . . . . .                                          | 7-2  |
| Activation . . . . .                                                          | 7-2  |
| Deactivation . . . . .                                                        | 7-3  |
| Connectability Status . . . . .                                               | 7-3  |
| Switched Virtual Circuits . . . . .                                           | 7-3  |
| Activation . . . . .                                                          | 7-3  |
| Activating Calls that Use SVCSC . . . . .                                     | 7-4  |
| Considerations . . . . .                                                      | 7-4  |
| Deactivation . . . . .                                                        | 7-4  |
| Deactivating Calls that Use SVCSC . . . . .                                   | 7-5  |
| Connectability Status . . . . .                                               | 7-5  |
| Application Attachment . . . . .                                              | 7-5  |
| Non-SNA Resources . . . . .                                                   | 7-5  |
| Permanent Virtual Circuits . . . . .                                          | 7-6  |
| Activation . . . . .                                                          | 7-6  |
| Deactivation . . . . .                                                        | 7-6  |
| Connectability Status . . . . .                                               | 7-6  |
| Switched Virtual Circuit . . . . .                                            | 7-6  |
| Activation . . . . .                                                          | 7-6  |
| Dial-In . . . . .                                                             | 7-6  |
| Dial-Out . . . . .                                                            | 7-6  |
| Connectability Status . . . . .                                               | 7-7  |
| Session Continuation . . . . .                                                | 7-7  |
| Application Attachment . . . . .                                              | 7-7  |
| GATE Considerations . . . . .                                                 | 7-7  |
| Fast Connect Considerations . . . . .                                         | 7-8  |
| FTPI Considerations . . . . .                                                 | 7-8  |
| X.21 Switched Connection Considerations . . . . .                             | 7-9  |
| DATE Considerations . . . . .                                                 | 7-9  |
| Activation . . . . .                                                          | 7-9  |
| Deactivation . . . . .                                                        | 7-10 |
| X.25 and SNA Resource Assignment Process . . . . .                            | 7-11 |
| Delayed Logical Link Activation (X.25 ODLC and NPSI V3R5 and Later) . . . . . | 7-11 |

This chapter describes the methods for controlling X.25 resources:

- SNA resources
- Non-SNA resources
- Delayed logical link activation (X.25 ODLC and NPSI V3R5 and later)

### SNA Resources

This section describes the operation of PVCs and SVCs. It also includes the required keywords, definitions, and activation and deactivation sequences, to permit safe and effective management of these resources.

### Permanent Virtual Circuits

PVCs are defined to the network as SNA dedicated resources. SNA resources, which operate on PVCs, are defined to VTAM and NCP in the same way they are defined to non-X.25 dedicated resources.

#### Activation

PVC activation for SNA resources starts when the MCH is activated. Once this occurs, the normal SNA activation sequence flows. Normal SNA flow includes ACTLINK, ACTPU, and ACTLU commands. SDLC commands are mapped to qualified logical link control (QLLC) packets for LLC type 3, and to PSH for LLC type 2.

To activate an SNA resource on a PVC, complete each activation step in the following order. To successfully complete the connection, this sequence must be followed carefully.

1. Activate the physical circuit line.

Activate the physical circuit line that is used by the virtual circuit to connect to the PSDN. The SNA activation sequence procedure requires that you activate the line first.

Physical circuit activation causes the voltage level in the data terminal ready (DTR) to raise. In response, the modem raises the DSR level, enabling the physical line to transmit and receive data.

2. Activate the physical circuit PU.

Activate the physical circuit PU to initiate the LAPB process. This activation causes the acknowledgments to flow and the physical circuit to reinitialize. An ACTPU command is transmitted to the MCH PU, completing this part of the activation sequence.

3. Activate the PVC line.

Activate the PVC (defined in the X25.LINE statement or the X25.VC statement) before or after MCH activation. No traffic occurs on the real line for this command. Though no traffic is actually transmitted in response to this command, you must adhere to the NCP hierarchy and activation sequence.

4. Activate the PVC PU.

Activate the physical unit associated with the PVC. The SNA request that activates the PVC also initiates the contact procedure. As a result, a PSCONT, QXID, or QSM command is transmitted to the remote DTE. A positive acknowledgment permits the contact procedure to be completed by a CONTACTED message returned to the initiator. An ACTPU command is transmitted to the PVC PU to complete the sequence.

5. Activate the PVC LUs.

The last part of this sequence is the activation of the LUs associated with the PVC. This sequence consists of an ACTLU command that is transmitted to the PVC LU. If the LU is successfully activated, a positive response is returned.



## Deactivation

VTAM and NCP definitions for SNA resources that operate on PVCs are similar to definitions related to normal NCP resources, and do not depend on whether you are using X.25. Like the activation sequence, the deactivation sequence closely resembles the SDLC resource sequence.

If the physical circuit is deactivated or has never been activated, the PVC will be unavailable for communication. If any of the PVC components (physical circuit, logical units, or physical units) are inactive, the subsequent components in the activation sequence will be unavailable. For example, if a physical circuit's physical unit is inactive, the PVC physical unit and logical unit cannot be activated.

The best method of deactivating the PVC is to reverse the order of the activation sequence. Reversing the order guarantees that all resources are returned to the correct state, and ensures that the network is synchronized and available for future communication. However, a PU can be deactivated if a RESET command is received from or sent to the network.

If for some reason a resource hangs in PENDING status, the network operator should issue a V NET,INACT,F command for the line related to the hung resource. In this case, every resource dependent on that line is deactivated.

## Connectability Status

When the PVC is available for communication, all resources associated with the SNA PVC indicate a status of ACTIVE. Check for the ACTIVE status to ensure that the PVC is ready for communication. The PVC is defined to NCP and VTAM as a dedicated SNA resource. As a result, the VTAM and NCP status for a PVC resource is displayed in the same way as any other dedicated SNA resource.

## Switched Virtual Circuits

SVCs are defined to VTAM and NCP as SNA switched resources. As a result, VTAM requires the creation of an SMN for SVCs.

### Activation

To activate an SNA resource on an SVC, complete each activation step in the following order. To successfully complete the connection, this sequence must be followed carefully.

1. Activate the physical circuit line.
2. Activate the physical circuit PU.
3. Activate the SVC line.

This can be done before MCH activation.

4. Activate the switched major node.

Activation of an SVC PU in an SMN results in the PU being ready to send or accept calls. The SMN physical unit completes the activation sequence with a connectable (CONCT) status.

Call-out is triggered by the application through the use of OPNDST OPTCD=ACQUIRE macro, or using an OPNDST OPTCD=ACCEPT combined with a SIMLOGON macro or VARY NET,LOGON= command.

## Controlling X.25 Resources

There is no logical unit activation with an SVC until the call is completed. In addition, the physical unit is not active until the call is completed.

Besides the normal activation sequence, the following SNA requests are also needed to activate SVCs:

- CONNOUT or ACTCONNIN
- REQCONT
- Multiple SETCVs
- RNAA

The activation process differs for call-in and call-out. See *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* for data flow diagrams.

### Activating Calls that Use SVCSC

For calls using SVCSC, the command `V NET,DIAL,ID=puname` of the SVCSC PU causes the call-out to be made. This PU in the SMN must have a PATH statement that contains the DIALNO keyword. The call can be made from either side when the calling PU has an associated PATH statement. There are no requirements on the called side for establishing the PU type 4 or 5 to PU type 4 or 5 session, besides activating the SMN.

If one of the NCPs fails during SHM disconnection but becomes active before reconnection, an initial call using `V NET,DIAL` must still be retransmitted. In this case, a normal SHM reconnection would fail and both sides would become inoperable. Before retransmitting the `V NET,DIAL` command, the side that remained active must be deactivated.

**Note:** X.25 ODLC does not support SHM.

After ANS and takeover of an NCP, the SVCSCs that were active must be reactivated from the new SSCP as all other resources. For example, this can be done using the `V NET,ACT,ID=ncpname,SCOPE=ALL` command. This allows the SSCP taking over to update its internal tables. Issuing a `V NET,DIAL` command before the activation previously noted for an SVCSC resource would desynchronize SSCP and X.25 status.

### Considerations

There is no link between the X.25 resource and SNA resources until the call is established. This structure eliminates MCH lockups that can occur because of a loss of synchronization between the SNA and the X.25 resources.

X.25 uses the SNA resources of a physical circuit as a pool of available resources. After obtaining the first available resource, X.25 creates a correspondence between the two sets of resources.

### Deactivation

The SVC deactivation sequence is a reversal of the activation sequence. SVC deactivation involves messages that flow between the PSDN, NCP, and VTAM. Message traffic results from the higher SNA levels that pass information to synchronize end points.

If the deactivation is initiated by a local or a remote DTE, the sequence can vary. The following three deactivation methods are used for SVC:

- Operator

- CLSDST command from application
- Clear packet to or from the network

The link between SNA and X.25 resources is broken at this point.

### Deactivating Calls that Use SVCSC

For calls using SVCSC, the command `V NET,HANGUP,ID=puname` of the SVCSC PU causes the call to be cleared. The call can be cleared from either side. Other possibilities, such as a CLEAR command from the network for usual error cases, can cause the call to be cleared. No SHM retries are executed then.

### Connectability Status

To allow for connection to a switched facility, every component must be in the correct state. The PU and LU in the SMN must be in a connectable (CONCT) state. The MCH line, MCH PU, and virtual circuit line must indicate an ACTIVE status.

The CONCT state is entered when the resource is available for connection. In this state, the VTAM resource is pending an activation sequence. Then the connection request is passed to the access method for final resolution of network address and resource pairings. The access method locates the appropriate physical unit in the SMN definition, and then determines the physical unit and link pairing.

After the activation sequence starts, the resource status changes to ACTIVE. This continues until the SVC logical unit is active with an LU-LU session setup. At this stage, the entire communications path is active, and the application session is established.

## Application Attachment

The application residing in the SNA host is attached to the SNA resource as if the PSDN did not exist. A normal BIND flows between the two logical units.

Determine the application considerations when using switched virtual circuits. For more information on the application requirements, see *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning*.

---

## Non-SNA Resources

Non-SNA resources operate similarly to SNA resources, except that:

- Non-SNA resources are defined and controlled by NPSI. NPSI takes the non-SNA resource and simulates it as an SNA resource to the host. Conversely, SNA requests sent from the host to non-SNA resources are handled by NPSI, rather than being sent directly to the non-SNA resources.
- Non-SNA resources, emulated as SNA resources by NPSI, are viewed as SNA LU type 1 devices. LU type 1 devices use an FID3 transmission header that is 2 bytes long and contains no sequence number. Consequently, no outstanding PIUs are allowed in the network because all PIUs must be acknowledged. Each PIU sent to a non-SNA resource is paced.

### Permanent Virtual Circuits

This section describes activation, deactivation, and connectability states for non-SNA PVCs.

#### Activation

PVC activation for non-SNA resources is similar to the activation sequence for SNA PVCs. The difference is that for non-SNA resources, NPSI, rather than the remote device, defines and controls the simulated SNA resources. To a contact command received for a PVC PU, a response is returned, but no DTE data (such as PSCONT or QSM) is forwarded to the PSDN.

#### Deactivation

PVC deactivation for non-SNA resources is also similar to the PVC deactivation sequence for SNA resources. Again, the difference is that NPSI defines and controls the simulated SNA resources rather than the remote device.

To a DISCONTACT command received for a PVC PU, a response is returned, but no DTE data (such as PSDISC or QDISC) is forwarded to the PSDN.

#### Connectability Status

VTAM displays the resource status as ACTIVE just as it does for an SNA PVC. However, this display only reflects the status as recorded by VTAM. SNA and non-SNA PVC resources appear the same to VTAM, even though the non-SNA remote device might not be ready.

### Switched Virtual Circuit

This section describes activation, deactivation, and connectability status for non-SNA SVCs.

#### Activation

SVC activation for both non-SNA and SNA resources is similar except that the SNA requests are not sent to the remote DTE. Instead, all SNA requests are processed by NPSI and the LU simulator. Synchronization between VTAM statuses and the remote DTE is ensured by the calling sequence.

#### Dial-In

The activation process for dial-in from a non-SNA resource is less complicated than for SNA resources, because only the Call Request and Call Accepted packets are transmitted across the network. Resource activation and session setup is performed between the NCP, NPSI, and VTAM.

#### Dial-Out

Dial-out to non-SNA resources follows the same process as that described for SNA resources. The difference is that fewer commands are processed across the network. Once the calling sequence is complete, PU and LU activation occurs as a series of command exchanges between the access method and the NPSI LU simulator.

### Connectability Status

Connectability status for SVCs, whether they are non-SNA resources or SNA resources, is the same.

### Session Continuation

For switched connections, the NCP allows a switched line and PU to remain active when the owning SSCP fails. With the owning SSCP inoperable, an alternate SSCP can acquire ownership of the resource. When the original SSCP becomes operable, it can regain ownership. Thus, when the session partners can still be accessed, the active LU-LU sessions belonging to that PU are not disturbed.

By coding ANS=CONT for a PU in the VTAM switched major node, the NCP allows session continuation.

### Application Attachment

To a host application, non-SNA resources appear to be defined as SNA resources. Actually, the host application is in session with the NPSI-simulated SNA resource. This resource is defined to the host application program as an SNA LU type 1 device.

NPSI takes the information transmitted to the simulated LU, prepares it for the PSDN, and transmits it. Therefore, the burden of support is transferred from the host application program to NPSI; however, the application must support the LU type 1 session.

### GATE Considerations

GATE processing requires that a session be established between the CTCP and the physical circuit LU before connection to the network resources requiring LLC type 4 can be completed. The MCH line, PU, and LU must be activated first; afterwards, the CTCP can establish a session with the MCH LU. The following processes are unique to GATE:

- All data for GATE processing passes between the CTCP and the virtual circuit LU. Consequently, the CTCP must either process the request or forward it to the appropriate application program.
- GATE usage is determined on a virtual circuit-by-virtual circuit basis. Therefore, a single physical circuit can support LLC type 4 and other types of LLCs.
- The MCH LU does not have to be active or in session for calls that do not require LLC type 4. The MCH LU to CTCP session is required only for LLC type 4 virtual circuits.
- GATE call-out requires no PATH statement in the switched major node, because NPSI handles this call-out as a simulated call-in request.
- The CTCP must issue CLSDST to deactivate the virtual circuit, because NPSI does not generate an INOP RU at clear time.

**Note:** If the CTCP is canceled or abends, or if the CTCP access method control block (ACB) is deactivated, the resources controlling the virtual circuits connected to this CTCP must be deactivated and then reactivated to avoid difficulties. In effect, the LUs that are undergoing logon processing at the time the CTCP fails stay active and are not reusable until the resources are deactivated or reactivated. If deactivating and reactivating the resources fails, the line resources associated

## Controlling X.25 Resources

with the hung virtual circuits must be deactivated with the VARY INACT,F form of the deactivate command, followed by activation.

## Fast Connect Considerations

If you plan to use fast connect, consider the following:

- Sessions between the CTCP and the virtual circuit LUs must be preestablished.
- NCP and VTAM operate as though they are communicating with nonswitched resources, while NPSI is controlling the SVCs. NPSI maps the SVCs to the nonswitched lines. The CTCP must provide the ability to display the status of the SVC if it is required for operation.

Activate the SNA resources before activating the MCH. This permits the sessions to be established prior to any connection requests.

Figure 7-1 shows how resources are allocated to avoid assignment collisions. This assignment process only applies to fast connect.

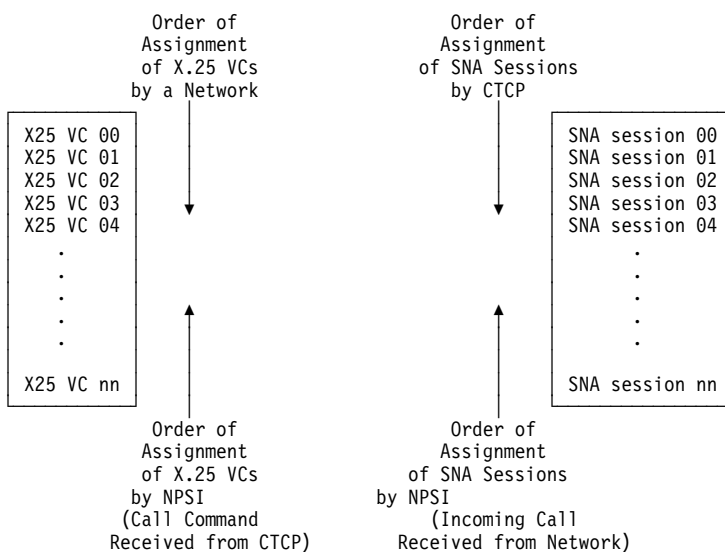


Figure 7-1. Virtual Circuit and SNA Resource Assignment Order (Fast Connect Only)

## FTPI Considerations

If you plan to use fast transaction processing interface (FTPI), consider the following:

- Sessions between the CTCP and the virtual circuit LUs must be preestablished.
- NCP and VTAM operate as though they are communicating with nonswitched resources, while NPSI is controlling the SVCs. NPSI maps the SVCs to the nonswitched lines.
- Sessions remain active even though a virtual circuit is not established. The CTCP must provide the ability to display the status of the SVC if it is required for operation.

Activate the SNA resources before activating the MCH. This permits the sessions to be established before any call-in requests.

## X.21 Switched Connection Considerations

If you plan to use an X.21 switched connection, you should consider the following:

- An X.21 switched connection can be used with GATE only.
- An MCH must be dedicated to the X.21 switched connection.

At link activation, NPSI uses the X21SW keyword on the X25.MCH statement to inform the scanner that the physical line is an X.21 switched line.

- A maximum of 15 digits can be reserved for X.21 dialing.

The X.21 dial digits, used to perform the X.21 outgoing call, are contained in the X.25 called DTE address as an NPSI call request packet built by CTCP and forwarded to GATE in a CALL REQUEST command. These dial digits represent the destination of the call, such as the number of the ISDN central exchange.

- X.21 incoming calls are accepted only when NPSI is in session with the CTCP (CTCP to MCH LU session).

When establishing a control session between CTCP and MCH LU, NPSI sets the physical interface in Monitor Incoming Call state so that X.21 incoming calls can be received on the X.21 switched line.

- At least one virtual circuit must be set in answer mode before any calls can be received.
- NPSI does not support X.21 calling or called line identification.
- NPSI does not support X.21 Short Hold Mode or X.21 Multiple Port Sharing.

## DATE Considerations

If DATE is used, the order and method of activation for virtual circuits is slightly modified. In general, DATE (and the associated CTCP) is used only for call setup and termination. During the application session, DATE is entirely out of the data flow. Unlike GATE, DATE is used for all virtual circuits on a physical circuit.

### Activation

To activate a virtual circuit for DATE, follow each of these steps in the correct order. To successfully complete the connection, follow this sequence carefully.

1. Activate the physical circuit.

Activate the physical circuit used by the virtual circuit that connects to the PSDN. Remember that the SNA activation sequence requires you to activate the physical line first.

Physical circuit activation causes the voltage level in the DTR to raise. In response, the modem raises the DSR level, enabling the physical line to transmit and receive data. You must reactivate the MCH line after CTCP deactivation or failure, because the MCH line is then inactive.

2. Activate the physical circuit PU.

During this step, nothing is sent to the network. The ACTPU is immediately completed.

3. Activate the MCH LU.

The LU is automatically activated because ISTATUS=ACTIVE is generated for the DATE MCH LU.

## Controlling X.25 Resources

4. Verify that the CTCP has established a type 1 session with the MCH LU.
5. Verify that the CTCP has sent the RESTART command to initiate the LAPB process. After the CTCP starts the line setup, it can monitor the link.

**Note:** If the CTCP is canceled or abends, or if the CTCP ACB is deactivated, the switched major node controlling the virtual circuits connected to this CTCP must be deactivated and then reactivated to avoid hung resources.

**PVC Setup:** DATE has very little effect on PVC activation. DATE is only involved when the Reset, Interrupt, and qualified packets are exchanged between the CTCP and NPSI. In this situation, DATE takes the appropriate packet information and passes it on to the CTCP.

**SVC Setup:** DATE has an effect on SVC activation. For example, the CTCP must determine when a user application program is ready to be called or is waiting to call out. Virtual circuit type, packet size, and window size must all be defined to the CTCP before it can pass that information on to NPSI.

For call-in, DATE processes the Call Request packet and forwards it to the CTCP. The CTCP then sends a CALL ACCEPTED command to NPSI that determines LLC type, window size, and packet size. Request Contact (REQCONT) is provided at this time.

For call-out, an operator command or an application-initiated message must be sent to the CTCP. Call-in is simulated to VTAM even though the CTCP can perform call-out. Therefore, the PATH statement is not needed in DATE's switched major node.

NPSI can build a logon for LLC types 0 and 5, if requested by the CTCP. For more details on DATE CTCP, see *NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning*.

### Deactivation

DATE session deactivation can occur in the following ways:

- NPSI receives a Clear Indication packet from the PSDN. Upon receipt of this packet, NPSI sends a CLEAR command to the CTCP. The CTCP sends a CLEAR CONFIRMATION command to NPSI, which in turn forwards the command to the network and generates an INOP to the virtual circuit for the host. This causes the application's LOSTERM exit to be invoked, permitting the session to be cleared when the CLSDST command is executed.
- The application issues a CLSDST command. Upon receipt of the resulting ABCONN request, NPSI sends an information report to the CTCP that sends a CLEAR command back to NPSI. Afterwards, NPSI forwards a Clear packet to the PSDN. Upon receipt of the Clear Confirmation packet from the PSDN, deactivation is complete.
- The CTCP can be deactivated normally by operator intervention, or abnormally by an abend. Once the CTCP is reactivated, the MCH link must also be reactivated and the session to the MCH LU re-established.
- The application requests the CTCP to clear the session by having the CTCP send a CLEAR command to NPSI. DATE translates this command into a Clear Request packet and forwards this information to the PSDN. A Clear Confirmation packet is returned to NPSI, triggering an INOP to the host. This causes



the application's LOSTERM exit to be invoked. Session termination occurs when the CLSDST command is executed.

- The CTCP fails for some reason and the MCH line is made inoperable. As a result, all dependent virtual circuits are cleared (SVCs) or reset (PVCs) either by the network, after a timeout of approximately two minutes, or by NPSI when the CTCP is restarted.

### X.25 and SNA Resource Assignment Process

Figure 7-2 shows the direction in which the resources are allocated and lists which program allocates each type of resource. Assignments are made in this order to avoid assignment collisions.

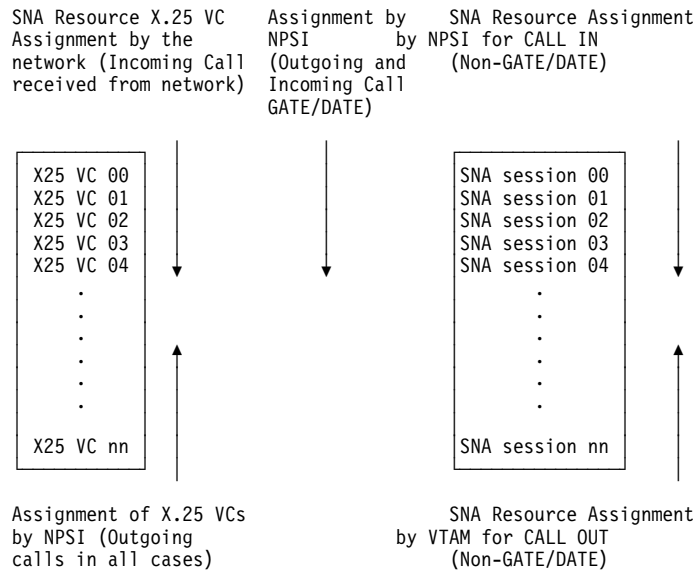


Figure 7-2. X.25 Virtual Circuit and SNA Resource Assignment

## Delayed Logical Link Activation (X.25 ODLC and NPSI V3R5 and Later)

VTAM establishes a hierarchical relationship between a physical resource (an MCH), and its logical (subordinate) resources (virtual circuits) when you code PHYSRSC=YES on the X25.MCH statement.

VTAM considers this physical resource and its subordinate logical resources as a complex link. In this case:

- VTAM automatically activates logical resources that it owns when the physical resource is activated without the need for a user-written CLIST or coordinated operator action.
- VTAM does not activate a logical resource subordinate to a physical resource that has not been previously activated.
- When displaying a logical resource, the name of the physical resource is also displayed.

**Note:** The previously described behavior only applies to logical resources owned by the same SSCP as the physical resource.



---

## Appendix A. Defining X.25 PSDN Facilities

|                                                               |      |
|---------------------------------------------------------------|------|
| Flow Control Parameter Negotiation                            | A-2  |
| Coding Requirements                                           | A-2  |
| Operation                                                     | A-3  |
| Called Line Address Modified Notification Facility            | A-3  |
| Priority of Data Facility (NPSI V3R4 and Later)               | A-4  |
| Called Address Extension Facility (AEF) (NPSI V3R4 and Later) | A-5  |
| Closed User Group                                             | A-5  |
| Coding Requirements                                           | A-5  |
| Operation                                                     | A-6  |
| Fast Select                                                   | A-6  |
| Coding Requirements                                           | A-6  |
| Operation                                                     | A-6  |
| High-Priority Class of Traffic                                | A-6  |
| Coding Requirements                                           | A-6  |
| Operation                                                     | A-7  |
| One-Way Logical Channel                                       | A-7  |
| Incoming Coding Requirements                                  | A-7  |
| Outgoing Coding Requirements                                  | A-7  |
| Reverse Charging                                              | A-7  |
| Coding Requirements                                           | A-8  |
| Operation                                                     | A-8  |
| Throughput Class Negotiation                                  | A-8  |
| Coding Requirements                                           | A-8  |
| Operation                                                     | A-9  |
| Optional Facilities Using X.25 Statements                     | A-9  |
| Virtual Circuits without a CTCP                               | A-9  |
| Virtual Circuits with a CTCP                                  | A-9  |
| Optional Facilities Table                                     | A-10 |

This appendix describes how X.25 supports various optional user facilities defined in Recommendation X.25. It describes the differences between virtual circuits with and without communication and transmission control programs (CTCPs) and describes the following optional facilities:

- Flow control parameter negotiation
- Called line address modified notification facility
- Priority of data facility
- AEF
- Closed user group
- Fast select
- High-priority class of traffic
- One-way logical channel
- Reverse charging
- Throughput class negotiation
- Optional facilities table

X.25 ODLC does not support the fast select facility.

---

### Flow Control Parameter Negotiation

X.25 supports flow control parameter negotiation. This optional facility allows packet and window size negotiation on a per call basis. When the calling DTE subscribes to flow control parameter negotiation, the DTE can request specific packet and window sizes for each direction of traffic flow. If a size is not specified, the default value is used.

### Coding Requirements

To code flow control parameter negotiation, use:

- Packet length:

Facility code field = X'42'  
Facility field parameter = '0x0y'

where:

x Is the packet size for transmission from the called DTE.  
y Is the packet size for transmission from the calling DTE.

| PACKET SIZE CODE | PACKET LENGTH |
|------------------|---------------|
| 4                | 16            |
| 5                | 32            |
| 6                | 64            |
| 7                | 128           |
| 8                | 256           |
| 9                | 512           |
| A                | 1024          |
| B                | 2048          |
| C                | 4096          |

- Window size:

Facility code field = X'43'  
Facility field parameter = 'wwzz'

where:

ww Is the window size for transmission from the called DTE.  
zz Is the window size for transmission from the calling DTE.

An example of a facility field is:

420909430404

where:

42 Is the flow control packet size negotiation facility.  
0909 Is the packet size.  
43 Is the flow control window size negotiation facility.  
0404 Is the window size for each direction.

This specification results in a packet size of 512 and a window size of 4 for each direction of traffic.

## Operation

X.25 checks the facility values in all the Incoming Call and Call Connected packets, and will accept the call only if *all* of the following three conditions are met:

1. x must be defined between X'4' and X'C'.
2. Negotiated packet size must be less than the maximum frame length.
3. ww must be less than or equal to 7 for modulo 8, or less than or equal to 127 for modulo 128.

In GATE and DATE, depending on the generation option, the flow control parameters included in the facility field of the Call Connected packet are taken into account by NPSI. The three conditions described previously must be met. If one of the conditions is not met, or if an error is found, the Call Connected packet is sent to the CTCP to clear the connection.

---

## Called Line Address Modified Notification Facility

Called line address modified notification is an optional user facility used by the DCE in the Call Connected or Clear Indication packets. It informs the calling DTE of the reasons the called DTE address in the packet differs from the called DTE address specified in the Call Request packet.

If the DTE has more than one address, the called address modified notification facility can be used by the DTE in the Clear Request or Call Accepted packets. This is done if the DTE address exists in the packet and is different than the DTE address specified in the Incoming Call packet.

Table A-1 illustrates the causes defined for this facility and their corresponding hexadecimal values.

Table A-1. Called Line Address Modified Notification

| Facility                                                                                         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|--------------------------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|
| Call redirection due to originally called DTE busy                                               | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Call Distribution within a hunt group                                                            | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Call redirection due to originally called DTE out of order                                       | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| Call redirection due to prior request from originally called DTE for systematic call redirection | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Call DTE originated <sup>1</sup>                                                                 | 1 | 0 | X | X | X | X | X | X |
| Call deflection by the originally called DTE busy <sup>2</sup>                                   | 1 | 1 | X | X | X | X | X | X |

**Note:**

<sup>1</sup> Each X can be independently set to 0 or 1 by the called DTE. Each X is passed transparently to the Calling DTE, Bit 8, when received from the called DTE. When bit 8 is not set to 1, it is forced to 1 by the DCE.

<sup>2</sup> The Xs are set by the originally called DTE in the Call Deflection Selection facility.

---

## Priority of Data Facility (NPSI V3R4 and Later)

The priority of data (QOS) facility is an optional facility defined to support OSI network services. QOS provides priority for the following functions:

- Priority of data on a connection
- Priority to gain a connection
- Priority to keep a connection

For NPSI, QOS enhances the internal process for queuing a packet before transmission to the network for multiple peripheral node traffic priorities.

The priority is also specified for what is targeted or available, and the minimum value that is acceptable.

Figure A-1 shows how NPSI uses the OSI priority of data facility for determining targeted, available, and minimum values.

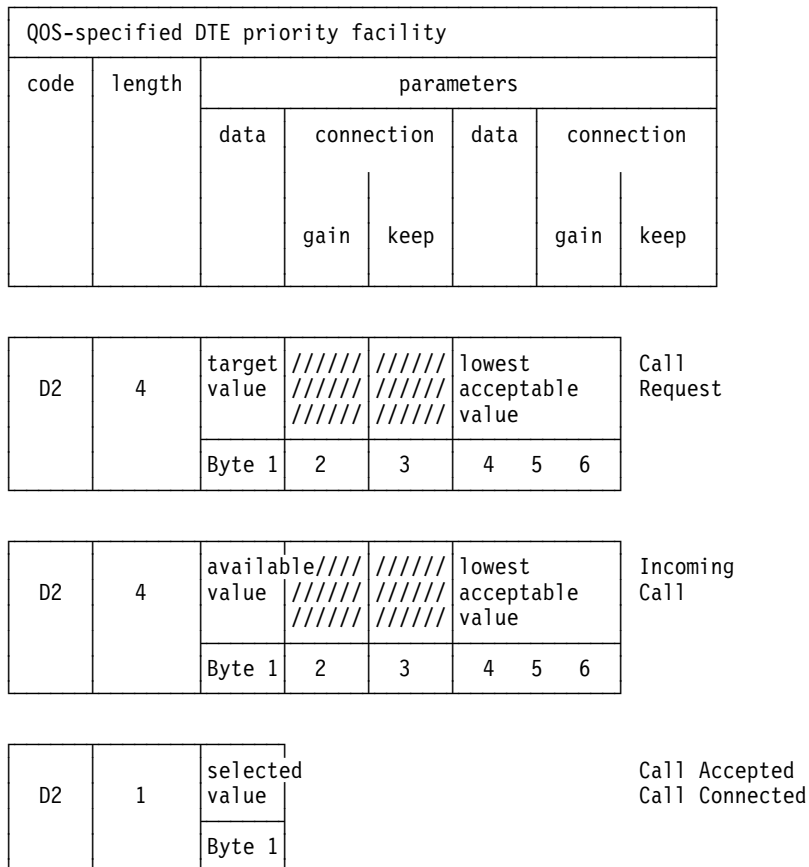


Figure A-1. NPSI Use of QOS Facility

Bytes 1, 2, and 3 of the facility field contain the following values, depending on the packet:

- |                       |                 |
|-----------------------|-----------------|
| <b>Call Request</b>   | Targeted value  |
| <b>Incoming Call</b>  | Available value |
| <b>Call Accepted</b>  | Selected value  |
| <b>Call Connected</b> | Selected value  |

Bytes 4, 5, and 6 of the facility field contain the minimum value that is acceptable for each of the functions.

NPSI uses only the first byte (the selected value) of the Call Accepted or Call Connected packet. Therefore, if this facility is used, the facility field must be at least 2 bytes long or the default priority is used for the call. All other bytes are not utilized by NPSI.

A CTCP can use the other bytes to determine whether the call should be cleared, or if the requested range of values can be used.

For calls whose values range from X'0F' (15) through X'FF' (255), the default value specified on the DEFPRI keyword on the X25.MCH statement is used.

---

## Called Address Extension Facility (AEF) (NPSI V3R4 and Later)

The Called Address Extension Facility (facility code X'C9') calls X.213/ISO 8348 addresses to establish routing decisions based on this facility.

The byte following the facility code (X'C9') is the length of the AEF parameter field in bytes. Table A-2 shows the meaning of the first two bits.

---

*Table A-2. Use of Called Address Extension in Bits 0 and 1*

| bit 0 | bit 1 | Use of called address extension                                                               |
|-------|-------|-----------------------------------------------------------------------------------------------|
| 0     | 0     | To carry a called address assigned according to Recommendation X.213/ISO 8348                 |
| 0     | 1     | Reserved                                                                                      |
| 1     | 0     | Other (to carry a called address not assigned according to Recommendation X.213/ISO 8348 AD2) |
| 1     | 1     | Reserved                                                                                      |

---



---

## Closed User Group

This option restricts a set of DTEs from access to another DTE set. DTEs belonging to one CUG can communicate with each other, but are denied communication with DTEs outside their group.

## Coding Requirements

To code CUG in the Call Request packet use:

Facility code field = X'03'

CUG number as defined to the PSDN

The index to the closed user group is in the form of 2-decimal digits. Each digit is binary coded. Bit 5 is the low-order bit of the first digit and bit 1 is the low-order bit of the second digit.

**Note:** Indexes to the same closed user group at different DTE to DCE interfaces, can differ.

The following example shows a request for closed user group 14:

0314

### Operation

X.25 does not perform any analysis on the content of the facility field, and accepts the call as long as none of the CUG keywords are specified in the RFAC keyword of the X25.NET statement of the corresponding network.

---

### Fast Select

The fast select option allows data to be transferred to or from the CTCP in a Call Request, Incoming Call, Call Accepted, Call Connected, Clear Request, or Clear Indication packet on an SVC within the CALL USER DATA and CLEAR USER DATA fields.

### Coding Requirements

To code the fast select option, use:

Facility code field = X'01'

Facility parameter field = X'80' (no restriction on response)

Facility parameter field = X'C0' (restriction on response); Call Accepted packets cannot be used

The following example shows the facility field requesting fast select:

```
01C0
```

### Operation

Communication between DTEs occurs through the use of:

- Incoming Call or Call Request packet
- Call Accepted or Call Connected packet
- Clear Request or Clear Indication packet

This contrasts to the normal mode of using Information packets as a communication method. The control packet's user data field is restricted in size to 128 bytes. This facility is only supported through DATE or GATE.

Fast select is an optional facility that can be rejected through the RFAC keyword in the X25.NET statement.

---

### High-Priority Class of Traffic

The high-priority class of traffic facility allows data to be transferred between the DTE and the DCE on an SVC with high-priority. This option restricts the requests and responses to 128 bytes in length with no support for the M bit.

### Coding Requirements

To code the high-priority class of traffic facility, use:

Facility code field = X'01'

Facility parameter field = X'00' (high-priority class of traffic not requested)

Facility parameter field = X'02' (high-priority class of traffic requested)



The following example shows the facility field requesting high-priority class of traffic:  
0102

## Operation

High-priority class of traffic is supported for outgoing and incoming calls. Packets must be 128 bytes long. For incoming calls, X.25 checks for high-priority service requests in the facility fields and internally updates the maximum packet size during the call. For outgoing calls, X.25 supports this facility by using the OPTFACL keyword of the X25.OUFT statement associated with a VCCPT entry containing a MAXPKTL parameter of 128.

In GATE and DATE, depending on the generation option, and if High Priority Class of Traffic is requested in the facility field of the Call Connected packet, this facility is taken into account by NPSI.

---

## One-Way Logical Channel

The one-way logical channel facility limits the use of a logical channel to either incoming or outgoing calls.

## Incoming Coding Requirements

The logical channel number of an incoming one-way logical channel must be *greater* than the greatest value for a permanent virtual circuit and *less* than the smallest value for a two-way virtual circuit. Thus, if LCNs 1—5 are used for PVCs and you want five logical channels for one-way incoming calls, these LCNs must start at a value greater than 5, but a lesser value than 4095.

## Outgoing Coding Requirements

The logical channel number of an outgoing one-way logical channel must be *greater* than the greatest value for a two-way virtual circuit but *less* than 4095. Thus, if LCNs 1—5 are used for two-way virtual circuits and you want five logical channels for one-way outgoing calls, you must assign LCNs greater than 5, but less than 4095, to these logical channels.

Restricted virtual circuits are defined through the use of the CALL keyword of the X25.LINE or X25.VC statement. The CALL keyword, which defines how an SVC is to be established, has the following options:

- CALL=IN (one-way in)
- CALL=OUT (one-way out)

---

## Reverse Charging

The reverse charging facility causes the PSDN to charge the called DTE rather than the calling DTE. If this facility is not specified, the calling DTE is assessed all connection charges.

### Coding Requirements

To code reverse charging in the Call Request packet, use:

Facility code field = X'01'

Facility parameter field = X'00' (reverse charging not requested)

Facility parameter field = X'01' (reverse charging requested)

The following example shows the facility field requesting reverse charging:

0101

### Operation

X.25 accepts the presence of the reverse charging facility field in an Incoming Call packet without analysis of the content unless REVCHG is specified with the RFAC keyword of the X25.NET statement. If REVCHG is specified, the call is cleared. Reverse charging can be requested in an outgoing call using the OUFT index in the PATH statement. Otherwise, reverse charging acceptance can be controlled through the use of DATE and GATE CTCPs.

---

### Throughput Class Negotiation

The throughput class of a virtual call defines the effective rate in which data can be transferred. Both packet size and window size affect this rate, but other factors such as the line speed also contribute.

Throughput class can only be negotiated *downward*. Downward negotiation reduces the effective data exchange rate and the associated transmission cost. This resulting data exchange rate effectively reduces the priority of a virtual circuit, which in turn reduces the use of the connecting physical circuit.

The PSDN forces the smallest throughput class value for each flow if there is a conflict between the throughput classes for transmission and the default for reception. This case applies only to switched virtual circuits.

### Coding Requirements

To code the throughput class negotiation facility use:

Facility code field = X'02'

Facility parameter field = bits 0–3

Facility parameter field = bits 4–7

The throughput class for transmission from the calling DTE is indicated in bits 4, 5, 6, and 7. The throughput class for transmission from the called DTE is indicated in bits 0, 1, 2, and 3. Bits 3 and 7 are the low-order bits of each throughput class indicator.

The following example shows the facility field requesting throughput class negotiation with 2400 bits per second for both the called and calling DTEs:

0277

## Operation

X.25 accepts the presence of the throughput class negotiation facility field in an Incoming Call packet without analysis of the content unless THRUPUTCL is specified with the RFAC keyword of the X25.NET statement. If THRUPUTCL is specified, the call is cleared.

---

## Optional Facilities Using X.25 Statements

The following section describes:

- Virtual circuits without a CTCP
- Virtual circuits with a CTCP

### Virtual Circuits without a CTCP

For virtual circuits without a CTCP, X.25 selects the flow control parameter values according to the following hierarchy:

- In the facility fields of the Incoming Call or Call Connected packets
- In the X25.VCCPT statement

**Note:** Facilities defined in the X25.OUFT statement are not taken into account. However, you must code the VCCPT facility entries to match the OUFT facility entries. In the PATH macro, xx and yy indexes must point to compatible entries.

### Virtual Circuits with a CTCP

For virtual circuits with a CTCP, the CTCP defines the flow control parameter values to NPSI in the CALL REQUEST or CALL ACCEPTED command. The values specified in the commands from the CTCP *should* correspond to either the values specified in the facility fields of the corresponding packet or the values that the network will use by default.

NPSI passes these facilities to the CTCP or to the PSDN without any processing. Negotiation occurs directly between the CTCP and the PSDN. If the facilities are not accepted by the CTCP or the PSDN, the call must be cleared by the CTCP or the PSDN.

NPSI does not check whether the packet and window sizes are the same in both directions. NPSI is transparent to all these negotiations. If the packet size and window size passed to the CTCP in the CALL CONNECTED command are different from the sizes specified to NPSI in the CALL REQUEST, the CTCP must clear the call.

**Note:** In GATE and DATE, if you code INTFAC=YES on the X25.MCH statement, the flow control parameters included in the facilities of the Call Connected packet are taken into account by NPSI. The following three facilities are analyzed:

- Packet length negotiation
- Window size negotiation
- High-priority class of traffic

Duplicate facilities are authorized (the last one is taken), regardless of the value coded for NETTYPE. It is the CTCP's responsibility to detect duplicate facilities.

### Optional Facilities Table

CCITT defines optional user facilities. If DATE or GATE is used, NPSI allows all facilities to be supported by the CTCP. However, if DATE or GATE is *not* used, NPSI supports the following facilities. Table A-3 and Table A-4 show NPSI support of CCITT facilities.

*Table A-3 (Page 1 of 2). NPSI Support*

| <b>Optional User Facility</b>             | <b>GATE/DATE</b> | <b>Other Option</b> |
|-------------------------------------------|------------------|---------------------|
| Bilateral CUG Related Facilities          | S                | S (out+GR)          |
| Call Redirection Notification             | S (by DCE)       | S (by DCE)          |
| Called Address Extension                  | S                | S (out+GR)          |
| Called Line Address Modified Notification | S                | S (GR)              |
| Called Redirection Notification           | S (by DCE)       | S (by DCE)          |
| Calling Address Extension                 | S                | S (out)             |
| Charging Information                      | S                | N/S                 |
| Closed User Group (CUG)                   | S                | S (out+GR)          |
| CUG Selection                             | S                | S (out+GR)          |
| CUG with Incoming Access                  | S                | S (out+GR)          |
| CUG with Outgoing Access                  | S                | S (out+GR)          |
| CUG with Outgoing Access Selection        | S                | S (out+GR)          |
| D-Bit Modification                        | N/S              | N/S                 |
| Default Throughput Class Assignment       | S                | S                   |
| End-to-End Transit Delay Negotiation      | S                | S (out)             |
| Extended Packet Sequence Numbering        | S                | S                   |
| Fast Select                               | S                | N/S                 |
| Fast Select Acceptance                    | S                | N/S                 |
| Flow Control Parameter Negotiation        | S                | S                   |
| Hunt Group                                | S (by DCE)       | S (by DCE)          |
| Incoming Calls Barred                     | S (by DCE)       | S (by DCE)          |
| Incoming Calls Barred within CUG          | S                | S (out)             |
| Local Charging Prevention                 | S (by DCE)       | S (by DCE)          |
| Minimum Throughput Class Negotiation      | S                | S (out)             |
| Network User Identification               | S                | S (out)             |
| Nonstandard Default Packet Sizes          | S                | S                   |
| Nonstandard Default Window Sizes          | S                | S                   |
| One-way Logical Channel Incoming          | S                | S                   |
| One-way Logical Channel Outgoing          | S                | S                   |
| Online Facility Registration              | N/S              | N/S                 |
| Outgoing Calls Barred                     | S (by DCE)       | S (by DCE)          |
| Outgoing Calls Barred within CUG          | S                | S (out)             |
| Packet Retransmission                     | N/S              | N/S                 |

Table A-3 (Page 2 of 2). NPSI Support

| Optional User Facility                 | GATE/DATE | Other Option |
|----------------------------------------|-----------|--------------|
| Reverse Charging                       | S         | S (out)      |
| Reverse Charging Acceptance            | S         | S            |
| RPOA Selection                         | S         | S (out)      |
| Throughput Class Negotiation           | S         | S            |
| Transit Delay Selection and Indication | S         | S (out)      |
| Quality of Service Negotiation         | S         | S (GR)       |

**Legend:**

- S** Supported by DTE, or usage on a per-call basis depends on operational requirements.
- N/S** Not supported.
- By DCE** The DTE is not involved in the processing of this facility.
- Out** NPSI can set some facilities in the outgoing Call Request packets without processing or checking these facilities when they are received in an Incoming Call or Call Connected packet.
- GR** Incoming call is rejected if the facility is included in the call-in, is not associated with the facility parameter, and is specified to be rejected in the RFAC keyword of the X25.NET statement.

Table A-4. Support for Implementation Requirements

| Implementation Requirement     | Support                        |
|--------------------------------|--------------------------------|
| Call User Data Field           | S                              |
| Concurrent sharing with SNA    | S (subaddressing, CUD)         |
| Extended Addressing            | S (with GATE/DATE only)        |
| Extended Facilities and Length | S                              |
| Fast Select                    | S (with GATE/DATE only)        |
| Interrupt Packet Control       | S                              |
| Reset Packet Control           | S                              |
| Use of D bit                   | S                              |
| Use of M bit                   | S (no mapping on SNA chaining) |

**Legend:**

- S** Supported



---

## Appendix B. System Generation Examples

|                                                                            |       |
|----------------------------------------------------------------------------|-------|
| Example A—NPSI Basic Functions . . . . .                                   | B-2   |
| NCP and NPSI Generation Input Including JCL . . . . .                      | B-3   |
| NPSI Link Edit Step . . . . .                                              | B-12  |
| NCP Input Produced by NDF (NPSI) . . . . .                                 | B-13  |
| VTAM Main Switched Major Node . . . . .                                    | B-22  |
| VTAM Switched Major Node for SNA Type 2.1 and SVCSC . . . . .              | B-27  |
| MODETAB for Password Protection and LU 6.2 . . . . .                       | B-31  |
| Example B—NPSI Default and User-Defined IDNUM and Remote Loading . . . . . | B-33  |
| NCP and NPSI Generation Input—Example B1 . . . . .                         | B-33  |
| NCP Input Produced by NDF (NPSI)—Example B1 . . . . .                      | B-41  |
| VTAM Switched Major Node—Example B1 . . . . .                              | B-55  |
| NCP and NPSI Generation Input—Example B2 . . . . .                         | B-60  |
| NCP Input Produced by NDF (NPSI)—Example B2 . . . . .                      | B-68  |
| Switched Major Node—Example B2 . . . . .                                   | B-82  |
| NCP and NPSI Connection Input—Example B3 . . . . .                         | B-87  |
| NCP Input Produced by NDF (NPSI) - Example B3 . . . . .                    | B-91  |
| Example C—NPSI NPM and FTPI (V3R4) . . . . .                               | B-96  |
| NCP and NPSI Generation Input with NPM and FTPI—EXAMPLE C . . . . .        | B-96  |
| Example D—NPSI HPTSS and Delayed Logical Link Activation . . . . .         | B-102 |
| NCP AND NPSI Generation input including JCL—Example D1 . . . . .           | B-102 |
| NCP Input Produced by NDF (NPSI)—Example D2 . . . . .                      | B-107 |
| Example E—X.25/ISDN Interworking (NPSI V3R6 and Later) . . . . .           | B-117 |
| Example F—Generation Information for X.25 ODLC . . . . .                   | B-125 |

This appendix contains examples of the coding you need to install NPSI or X.25 ODLC in the NCP. Use this appendix when planning your installation process, but remember all network configurations differ.

The appendix is divided into six parts:

- Example A for NPSI basic functions
- Example B for NPSI default and user-defined IDNUM and remote loading
- Example C for NPSI NPM and FTPI
- Example D for NPSI HPTSS and delayed logical link activation
- Example E for NPSI X.25/ISDN interworking
- Example F for NCP X.25 ODLC

### Example A—NPSI Basic Functions

This example consists of six sections:

- NCP and NPSI generation input including JCL  
This section contains examples of NPSI features, such as different types of MCHs and virtual circuits.
- LKEDT (link edit) step  
This section contains an example illustrating how the LKEDT (link edit) step enables easier maintenance of NPSI if relinking of modules is all that is required.
- NCP input produced by NDF (NPSI)  
This section contains examples of NCP statements produced by the NPSI part of generation for input to VTAM. Also, examples are included of the default NPSI naming conventions and of the order of the statements generated for SVCs.
- Main switched major node  
This example illustrates some of the IDNUM usages that go with the NPSI generation.
- Switched major node for SNA type 2.1 and SVCSC  
This section contains examples of SMN entries to be used for SNA type 2.1 and SVCSC.
- MODETAB for password protection and LU type 6.2  
This section contains examples of mode tables for password protection and LU type 6.2.

**Note:** This is not intended to be a production example.



## NCP and NPSI Generation Input Including JCL

This section contains examples of NPSI features, such as different types of MCHs and virtual circuits.

```
//S1 EXEC PGM=IEHPRGM
//* CLEAN UP PREALLOCATED PDS USED TO SAVE CONTROL BLOCK OBJECT CODE
//SYSPRINT DD SYSOUT=A
//DD1 DD VOL=SER=MVSWK1,UNIT=3350,DISP=SHR
//SYSIN DD *
 SCRATCH DSN=X25.NCPV32.SA324GN,VOL=3350=MVSWK1,MEMBER=ICNTABL1
 SCRATCH DSN=X25.NCPV32.SA324GN,VOL=3350=MVSWK1,MEMBER=ICNTABL2
 SCRATCH DSN=X25.NCPV32.SA324GN,VOL=3350=MVSWK1,MEMBER=SA324LE
/*
//S2 EXEC PGM=IEBCOPY,ADDRSPC=REAL,REGION=128K
//* COMPRESS THE PREALLOCATED PDS
//SYSPRINT DD SYSOUT=A
//DISK DD DISP=SHR,DSN=X25.NCPV32.SA324GN
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(10,1),,CONTIG)
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(10,1),,CONTIG)
//SYSIN DD *
 COPY OUTDD=DISK,INDD=DISK
/*
//STEP1 EXEC PGM=ICNRTNDF,REGION=6000K,PARM='LINECNT=45'
//STEPLIB DD DSN=SYS1.SSPLIB,DISP=SHR
// DD DSN=SYS1.NPSILNK,DISP=SHR
//GENDECK DD *

*
* EXAMPLE OF GENERATION OF X.25 NPSI WITHIN NCP *
*
* - MCH7 (SL7) PVC 0 FOR PAD FOR USE WITH TSO *
* SVC 1 TO 7 FOR PCNE,PSH,PADI,BNNQLLC,INNQLLC *
* 2 SVC RESERVED FOR INN SVC(WITH 2 INN STATIONS)*
* - MCH0 (SL0) PVC 0 FOR PSH *
* SVC1,2,3,4 FOR PCNE,PSH,PADX,BNNQLLC *
* 1/IN 2,3/INOUT 4/OUT *
*
* - MCH1 (SL1) PVC 0 FOR PCNE *
* PVC 1 FOR INN (WE ARE SECONDARY) *
* PVC 2 FOR INN (WE ARE PRIMARY) *
* PVC 3 FOR BNN QLLC *
* PVC 4 FOR BNN QLLC PU T2.1 *
* PVC 5 FOR BNN QLLC PU T2.1 & LU6.2 *
* SVC 100 FOR GATE SUBADDRESSING(ANY LLC SELECTED)
*
* - MCH2 (SL2) PVC 0 FOR PCNE *
* SVC 1,2 FOR PCNE,PSH,PADI,BNNQLLC,GATE *
* 1/2 INOUT NULL CUD0 WILL SELECT PAD *
*
* - MCH3 (SL3) SVC 1,2 FOR PCNE,PSH,PADX,BNNQLLC (DATE) *
*
*
* - MCH4 (SL4) SVC 1,2,3,4,5,6,7 FOR FAST CONNECT *
* CONNECT=SUBD: CTCP SELECTED VIA SUBADDRESSING*
*

*
```

## Example A—NPSI Basic Functions

```

**
**
 OPTIONS USERGEN=X25NPSI, *INDICATE NEO NAME X
 NEWDEFN=(YES,,SUPP) *INCLUDE NPSI STMTS IN VTAMLST
BAKLOCAL PCCU DUMPDS=DUMPX25,SUBAREA=1
OTHLOCAL PCCU DUMPDS=DUMPX25,SUBAREA=16
REMTAM PCCU DUMPDS=DUMPX25,SUBAREA=2
*
BAKBUILD BUILD NPA=YES, X
 VERSION=V5R2, * NCP VERSION X
 LOADLIB=LNCP11, X
 TYPGEN=NCP, X
 MODEL=3745, X
 USGTIER=3, * NCP USGTIER X
 X25.USGTIER=3, * X.25 NPSI USGTIER X
 NETID=NETID1, * NETWORK ID X
 MAXSUBA=63, X
 SUBAREA=4, X
 BFRS=124, * BEST FOR X.25 NPSI PERFORMANCE X
 LTRACE=4, * ALLOWS LINE TRACE ON TWO MCH'S X
 OLT=YES, X
 DR3270=NO, X
 BRANCH=8000, X
 TRACE=(YES,100), * 100 ENTRIES IN ADDRESS TRACE TBL X
 NEWNAME=SA324GN, XX
 X25.PREFIX=S, * ALL NAMES START WITH S X
 X25.IDNUMH=02, * X25.IDNUMH MUST MATCH WITH SWITCHX
 TYP SYS=OS, * MAJOR NODE IDS.(2FIRST DIGITS) X
 X25.SNAP=YES, * OPTIONAL DIAGNOSIS SNAP INCLUDED X
 X25.MCHCNT=6, * NUMBER OF PHYSICAL LINKS X
 X25.MAXPIU=64K
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD, X
 RIMM,NAKLIM,SESSION,SSPAUSE,XMTLMT)
BAKHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=1, X
 DELAY=0.1,TIMEOUT=30
OTHHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16, X
 DELAY=0.1,TIMEOUT=30
REMHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=2, X
 DELAY=0.1,TIMEOUT=30
PATH14 PATH DESTSA=(1),ER0=(1,1),ER1=(1,1),ER2=(1,1) TO LOCAL
PATH116 PATH DESTSA=(16),ER0=(16,1),ER1=(16,1),ER2=(16,1) TO OTHLOCAL
PATH42 PATH DESTSA=(5,2,3),ER0=(5,2),ER1=(5,3),ER2=(3,4)
 LUDRPOOL NUMTYP1=10, * FOR NON SNA CALLS X
 NUMTYP2=10, *FOR QLLC OR PSH CALLS X
 NUMILU=2 *FOR LU6.2 SVC'S

 EJECT * AT LEAST ONE SDLC LINE
SDLC6 GROUP LNCTL=SDLC, * MUST BE DEFINED X
 TYPE=NCP, X
 DIAL=YES, X
 REPLYTO=20

SL6 LINE ADDRESS=6, * SDLC LINE FOR NIA FRONTAL X
 CLOCKNG=EXT, * X
 NEWSYNC=NO, X
 DUPLEX=HALF, X
 NRZI=NO, X
 CALL=INOUT, X

```

**Example A—NPSI Basic Functions**

```

RETRIES=(7,0,0)
SP6 PU ISTATUS=INACTIVE, X
 PUTYPE=2

 EJECT
SDLC5 GROUP LNCTL=SDLC, X
 TYPE=NCP, X
 DIAL=NO, X
 REPLYTO=3, X
 DISCNT=NO, X
 ISTATUS=INACTIVE

SL5 LINE ADDRESS=5, * LEASED SDLC MULTIPOINT LINE X
 CLOCKNG=EXT, X
 NEWSYNC=NO, X
 DUPLEX=HALF, X
 NRZI=NO,NPACOLL=YES, X
 RETRIES=(3)
 SERVICE ORDER=(SP5)
SP5 PU ADDR=C2, * ADDRESS 3274 X
 MAXDATA=261, X
 MAXOUT=3, X
 PUTYPE=2,NPACOLL=YES, X
 VPACING=2, X
 PACING=1
SU5 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,NPACOLL=YES
 EJECT

GRNPA GROUP LNCTL=SDLC, * FOR NPM RECORDING X
 TYPE=NCP, X
 DIAL=NO, X
 DISCNT=NO, X
 VIRTUAL=YES, X
 NPARSC=YES, X
 ISTATUS=INACTIVE
LNNPA LINE LINEFVT=NPAVFVT
PUNPA PU PUTYPE=2,PUFVT=NPAPFVT
LUNPA LU MAXCOLL=7,LUFVT=(NPALFVT,NPALFVT), X
 VPACING=0,LOCADDR=1

 EJECT
***** X25 NPSI INPUT
* ALLOW 3 VCCPT AND OUFT ENTRIES
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES
*
 X25.VCCPT INDEX=1, X
 MAXPKTL=128, X
 VWINDOW=1, * PACKET LEVEL WINDOW X
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.VCCPT INDEX=2, X
 MAXPKTL=128, X
 VWINDOW=2, X
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.VCCPT INDEX=3, X
 MAXPKTL=128, X
 VWINDOW=3, X

```

## Example A—NPSI Basic Functions

```

 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2, * PKT SIZE=128,VWINDOW=2 X
 OPTFACL=420707430202
*
X25.OUFT INDEX=3, * PKT SIZE=128,VWINDOW=3, AND X
 OPTFACL=03430102420707430303, * REVERSECHARGING REQUEST X
 USRFILD=1234567890

*
 EJECT

* MCH SL7: NO DATE/GATE BUT SVC INN

X25.MCH ADDRESS=7, *CONTROLLER LINE ADDR X
 ANS=CONT, X
 LCGDEF=(0,7), *MAX LCN IS 7 ON LCGN 0 X
 FRMLGTH=131, X
 MWINDOW=7, *LAP WINDOW X
 STATION=DTE, X
 TPTIMER=3, *LAPB RECOVERY PARAMETERS X
 TDTIMER=1, * X
 NPRETRY=7, * X
 NDRETRY=1, * X
 LLCLIST=(LLC0,LLC2,LLC3,LLC5), X
 LCN0=USED, X
 SPAN=X2501, *FOR NETVIEW X
 DBIT=NO, *NO DBIT SUPPORT ON THIS MCH X
 PAD=INTEG,TRAN=EVEN, *ALLOW INTEGRATED PAD X
 PWPROT=YES, *PASSWORD PROTECTION ON IPAD X
 ITRACE=NO, * LAP INTERNAL TRACE INACTIVE X
 SPEED=9600, *REAL MCH SPEED X
 TITIMER=1, *REAL T1 OF THE DCE X
 SVCINN=2, *NB OF SIMULT.SVC INN CONNECTIONX
 SHM=YES, *SHORT HOLD MODE ON SVCSC'S X
 SDRTCNT=3, *NB OF RETRIES X
 SDRTIME=10 *TIMER BETWEEN CALL RETRIES
*
X25.LCG LCGN=0 LOGICAL CHANNEL GROUP
*
*
* PVC FOR INTEGRATED PAD FOR USE WITH TSO AND WITH PASSWORD PROTECTION
*
X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC5,SPAN=X2501, *
 MAXDATA=2000, *SEGMENTING NOT SUPPORTED FOR PAD *
 SSCPFM=USSNT0, *LOGON ENDS WITH CRLF X
 MODETAB=MODTWX, *MODETAB FOR LU1 3767/TWX X
 DLOGMOD=TWXCONT, *FLIPFLOP OR CONTENTION PROT X
 TERM=TWX, *MUST BE DEFINED FOR TWX/3101 X
 USSTAB=USSTWX *SPECIAL USSTAB FOR TWX/3101
*
*** 2 LINE'S FOR SUBAREA DIAL(2 SVC'S RESERVED FOR SVC INN)
*
X25.LINE DSTNODE=INN, *DEFINE A SWITCHED INN VC X
 TYPE=S

```

## Example A—NPSI Basic Functions

```

X25.PU PUTYPE=4, ISTATUS=INACTIVE
*
X25.LINE DSTNODE=INN, *DEFINE A SWITCHED INN VC X
 TYPE=S
X25.PU PUTYPE=4, ISTATUS=INACTIVE
*
*
* DEFINE 7 SVC TO BE SHARED BETWEEN INN SVC'S (MAX OF 2 AT A TIME)
* AND BNN SVC (MAX OF 5 AT A TIME)
*
 X25.VC LCN=(1,7), VCCINDX=3, TYPE=S, CALL=INOUT, OUFINDX=1, X
 SPAN=X2501
*
 EJECT

* MCH SL0: NO DATE/GATE BUT TRANSPARENT PAD

 X25.MCH ADDRESS=0, *CONTROLLER LINE ADDR X
 ANS=CONT, X
 LCGDEF=(0,4), *MAX LCN IS 4 ON LCGN 0 X
 FRMLGTH=131, X
 MWINDOW=2, *LAP WINDOW X
 STATION=DTE, *FOR OWNERSHIP RECOVERY VIA X
 XMONLNK=YES, *INN SVC (INITIAL ACT IMPOSSIBLE) X
 TPTIMER=3, *LAPB RECOVERY PARAMETERS X
 TDTIMER=1, * X
 NPRETRY=7, * X
 NDRETRY=1, * X
 LLCLIST=(LLC0, LLC2, LLC3, LLC5), X
 LCN0=USED, X
 SPAN=X2501, *FOR NETVIEW X
 DBIT=NO, *NO DBIT SUPPORT ON THIS MCH X
 PAD=TRANSP, TRAN=EVEN, *ALLOW TRANSPARENT PAD X
 ITRACE=YES, *LAP INTERNAL TRACE ACTIVE X
 SPEED=9600, *REAL MCH SPEED X
 T1TIMER=1 *REAL T1 OF THE DCE
*
 X25.LCG LCGN=0
*
*** X25 LINE FOR PSH (PVC) LINE NAME=SL0000 PU=SP0000 LU=SU0000
*
 X25.LINE LCN=0, VCCINDX=3, LLC=LLC2, TYPE=P, SPAN=X2501
*
 X25.PU PUTYPE=2, ADDR=01, MAXDATA=265, VPACING=2, X
 PACING=2, ISTATUS=INACTIVE
 X25.LU LOCADDR=2, USSTAB=USSTD3, MODETAB=MODTD3
*
* 4 SVC'S: 1 IN, 2 IN/OUT, 1 OUT
*
 X25.VC LCN=1, VCCINDX=2, TYPE=S, CALL=IN, OUFINDX=2, SPAN=X2501
 X25.VC LCN=(2,3), VCCINDX=3, TYPE=S, CALL=INOUT, OUFINDX=3, X
 SPAN=X2501
 X25.VC LCN=4, VCCINDX=3, TYPE=S, CALL=OUT, OUFINDX=1, SPAN=X2501
*

* MCH SL1: GATE WITH SUBADDRESSING & PU T2.1

 X25.MCH ADDRESS=1, X

```

## Example A—NPSI Basic Functions

```

ANS=CONT, *MAX LCN IS 5 ON LCGN 0 X
LCGDEF=((0,5),(1,0)), *MAX LCN IS 0 ON LCGN 1 X
FRMLGTH=131, X
MWINDOW=3, LAP WINDOW X
STATION=DTE, X
XMONLNK=YES, *FOR EVENTUAL ACT VIA INN PVC X
SHUTD=NOINVCLR, *FOR IPAD CLSDST PASS X
TPTIMER=2, *LAPB RECOVERY PARAMETERS X
TDTIMER=3, X
NPRETRY=3, X
NDRETRY=3, X
LCN0=USED, X
PAD=INTEG,TRAN=EVEN, *ALLOW INTEGRATED PAD ON MCH X
GATE=GENERAL,SUBADDR=YES, X
LLCLIST=(LLC0,LLC2,LLC3,LLC4,LLC5), X
LLC0=(0,2),LLC2=8,LLC3=9, *SUBADDRESSING DIGITS USED TOX
LLC4=(1,4,3,7),LLC5=(6,5), *SELECT THE 5 LLC'S X
SPAN=X2501, *USE DEFAULT T1TIMER= 1SEC X
DBIT=NO *USE DEFAULT SPEED=9600BPS X
*
*
X25.LCG LCGN=0 *FIRST LOGICAL CHANNEL GROUP
*
* PVC FOR PCNE: LINK NAME=SL1000 PU=SP1000 LU=SU1000
*
X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC0,SPAN=X2501, *
MAXDATA=2000 *SEGMENTING NOT SUPPORTED FOR PCNE
*
***** X25 LINE FOR INN ***THIS SIDE IS SECONDARY*****
*
SL1001 X25.LINE LCN=1, X
VCCINDX=3, X
MONLINK=YES, X
DSTNODE=INN, *MANDATORY FOR INN X
LLC=LLC3, X
SPAN=X2501, X
TYPE=P
SP1001 X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE, X
SPAN=2501,
*
***** X25 LINE FOR INN *****THIS SIDE IS PRIMARY*****
*
SL1002 X25.LINE LCN=2, X
VCCINDX=3, X
MONLINK=YES, X
DSTNODE=INN, *MANDATORY FOR INN X
LLC=LLC3, X
SPAN=X2501, X
TYPE=P
SP1002 X25.PU PUTYPE=4,TGN=4,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE, X
SPAN=2501,
*
***** X25 LINE FOR BNN QLLC *****
*
SL1003 X25.LINE LCN=3, X
VCCINDX=3, X
LLC=LLC3, *DSTNODE=BNN DEFAULTED X
SPAN=X2501, X

```

## Example A—NPSI Basic Functions

```

 TYPE=P
SP1003 X25.PU PUTYPE=2,ADDR=01,MAXDATA=265,VPACING=2, X
 PACING=1,SPAN=X2501,ISTATUS=INACTIVE
SU1003 X25.LU LOCADDR=2,USSTAB=USSTD3,MODETAB=MODTD3
*
* *** PVC 4 FOR PU2.1 ON BNN QLLC ***
* *** WITH DEPENDENT LU'S ***
*
 X25.LINE LCN=4, X
 VCCINDX=3, X
 LLC=LLC3, * DSTNODE=BNN DEFAULTED X
 SPAN=X2501, X
 TYPE=P
*
 X25.PU PUTYPE=2,ADDR=01,VPACING=2,XID=YES,DATMODE=HALF, X
 PACING=1,SPAN=X2501,ISTATUS=INACTIVE,ANS=CONT
*
 X25.LU LOCADDR=2
*

*
* *** PVC 5 FOR PU2.1 ON BNN QLLC ***
* *** WITH INDEPENDENT LU'S ***
*
 X25.LINE LCN=5, X
 VCCINDX=3, X
 LLC=LLC3, * DSTNODE=BNN DEFAULTED X
 SPAN=X2501, X
 TYPE=P
*
 X25.PU PUTYPE=2,ADDR=01,VPACING=2,XID=YES,DATMODE=HALF, X
 PACING=1,SPAN=X2501,ISTATUS=INACTIVE,ANS=CONT
*
 X25.LU LOCADDR=0, * FOR INDEPENDENT LU'S X
 MODETAB=LENMODE
*
*
** VC USING SUBADDRESSING TO SELECT THE LLC TYPE *****
*
 X25.LCG LCGN=1 *SECOND LOGICAL CHANNEL GROUP
**
 X25.VC LCN=0, *SVC FOR SUBADDRESSING X
 VCCINDX=3, X
 OUFINDX=1, X
 CALL=IN, *USED FOR CALLIN ONLY X
 SPAN=X2501, X
 TYPE=SWITCHED

* MCH SL2 : GATE WITH USER DEFINED RELATION BETWEEN
* FIRST CUD BYTE AND CTCP/LLC

 X25.MCH ADDRESS=2, X
 ANS=CONT, X
 LCGDEF=(0,2), X
 FRMLGTH=131, X
 MWINDOW=3, LAP WINDOW X
 STATION=DTE, X

```

## Example A—NPSI Basic Functions

```

TPTIMER=3.1, X
TDTIMER=2, X
NPRETRY=7, X
NDRETRY=1, X
LCN0=USED, X
PAD=INTEG,TRAN=EVEN, X
GATE=GENERAL, X
SPAN=X2501, X
LLCLIST=(LLC0,LLC2,LLC3,LLC4,LLC5), X
CUD0=(00,02,21,22,NULL,03,B0,C1,C5,CC,FF), X
CTCP=(00,80,01,02,85,85,85,80,85,80,85)

*
* 3 CTCP'S FOR GATE.CUD0 SELECTIONS: CTCP0=00,CTCP1=21,CTCP2=22
* PCNE WILL BE SELECTED BY CUD0=02,C1,CC
* PAD WILL BE SELECTED BY CUD0=NULL,03,B0,C5,FF
*
* FOR CALLOUT IF L=7(PAD) THE CALL OUT PKT WILL HAVE A NULL CUD
* FOR CALLOUT IF L=(PCNE) THE CALL OUT PKT WILL HAVE 02 IN CUD0

X25.LCG LCGN=0
*
*** PVC FOR PCNE *****
*
X25.VC LCN=0, X
VCCINDX=2, X
LLC=LLC0, X
MAXDATA=2000, * SEGMENTING NOT SUPPORTED X
TYPE=P
*
** 2 SVC'S *****
*
X25.VC LCN=(1,2), X
VCCINDX=3, X
OUFINDX=1, X
CALL=INOUT, X
TYPE=SWITCHED

* MCH SL3: DATE MCH

*
X25.MCH ADDRESS=3, X
ANS=CONT, X
GATE=DEDICAT, X
LCGDEF=(0,2), *MAX LCN IS 2 ON LCGN 0 X
FRMLGTH=131, X
MWINDOW=7, LAP WINDOW X
STATION=DTE, X
TPTIMER=3, X
TDTIMER=1, X
NPRETRY=7, X
NDRETRY=1, X
LLCLIST=(LLC0,LLC2,LLC3,LLC5), X
LCN0=NOTUSED, X
DBIT=NO, X
SPEED=9600, X
TITIMER=1, X

```



## Example A—NPSI Basic Functions

```

 PAD=TRANSP,TRAN=EVEN *PADI NOT ALLOWED ON DATE MCH
*
 X25.LCG LCGN=0

*
 X25.VC LCN=(1,2),VCCINDX=2,TYPE=S,CALL=INOUT,OUFINDX=2
 EJECT
*

* MCH XL11: 7 VC'S FOR FAST CONNECT WITH 2 CTCP'S

*

XL12 X25.MCH ADDRESS=4,PUNAME=XP12,LUNAME=(XU12,XU22), X
 ANS=CONT, X
 LCGDEF=(0,7), X
 FRMLGTH=131, X
 MWINDOW=7, LAP WINDOW X
 STATION=DTE, X
 TPTIMER=3, X
 TDTIMER=1, X
 SPAN=X2501, X
 NPRETRY=7, X
 NDRETRY=3, X
 GATE=GENERAL, * NEEDED FOR FASTCONNECT X
 LLCLIST=LLC4, X
 LCN0=NOTUSED, X
 ITRACE=NO, *TO REDUCE PATH LENGTH X
 LOGAPPL=(GBGTPLS,GBGTPLT), *CTCP'S PLU NAMES X
 CONNECT=SUBD, * 2 CTCP'S SELECTABLE VIA THE X
 SUBD=(0,9,1), * LAST SUBADDRESSING DIGIT X
 CTCP=(0,1,1) *0 FOR CTCP 0, 9 AND 1 FOR CTCP 1
*
 X25.LCG LCGN=0 * DEFINE THE NETWORK'S VC'S
*
 X25.VC LCN=(1,7),VCCINDX=3,TYPE=S,CALL=INOUT
*
* DEFINE THE MAX NB OF LU'S USABLE AT THE SAME TIME ON CTCP0
*
 X25.FCG QTY=(5),CTCPNO=(0),SPAN=X2501,MAXDATA=2500, X
 PRFLINE=XL12,PRFPU=XP12,PRFLU=XU21,SUFFIX=0001
*
* DEFINE THE MAX NB OF LU'S USABLE AT THE SAME TIME ON CTCP1
*
 X25.FCG QTY=(5),CTCPNO=(1),SPAN=X2501,MAXDATA=2500, X
 PRFLINE=XL22,PRFPU=XP22,PRFLU=XU22,SUFFIX=0001
*

 X25.END

CAGROUP GROUP LNCTL=CA, X
 CA=TYPE6, X
 DELAY=0.0, X
 NCPCA=ACTIVE
CALINE1 LINE ADDRESS=08 TO LOAD ON 540=CCUA
CA1PU PU PUTYPE=5
 EJECT
 GENEND

```

## Example A—NPSI Basic Functions

```
END

//DBWORKFL DD UNIT=SYSDA,SPACE=(TRK,10)
//SYSPRINT DD SYSOUT=A
//PRINTER DD SYSOUT=A
//TBL1SRCE DD DSN=&&SRCE1,DISP=(,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,10)),DCB=BLKSIZE=3200
//TBL1LIST DD SYSOUT=Z
//*TBL1OBJ DD DUMMY TABLE 1 OBJECT FILE
//TBL1OBJ DD DSN=X25.NCPV32.SA324GN(ICNTABL1),DISP=OLD,
// VOL=SER=MVSWK1,UNIT=3350
//TBL2SRCE DD DSN=&&SRCE2,DISP=(,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,10)),DCB=BLKSIZE=3200
//TBL2LIST DD SYSOUT=Z
//*TBL2OBJ DD DUMMY TABLE 2 OBJECT FILE
//TBL2OBJ DD DSN=X25.NCPV32.SA324GN(ICNTABL2),DISP=OLD,
// VOL=SER=MVSWK1,UNIT=3350
//NEWDEFN DD DISP=SHR,DSN=SYS1.VTAMLST(SA324GN)
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10)),DISP=(,DELETE)
//SYSLIB DD DSN=SYS1.SNCPMAC1,DISP=SHR
// DD DSN=SYS1.SBALMAC1,DISP=SHR
//LNKSTMT DD DSN=X25.NCPV32.SA324GN(SA324LE),DISP=OLD
/*
```

## NPSI Link Edit Step

This section contains an example illustrating how the LKEDT (link edit) step enables easier maintenance of NPSI if relinking of modules is all that is required.

```
//LKEDT EXEC PGM=IEWL,REGION=400K,
// PARM='LIST,NCAL,NOXREF,SIZE=(374K,48K),MAP'
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,DELETE)
//SYSLIN DD DSN=X25.NCPV32.SA324GN(SA324LE),DISP=OLD
//SYSPUNCH DD DSN=X25.NCPV32.SA324GN,DISP=OLD,UNIT=3350,VOL=SER=MVSWK1
//SYSPRINT DD SYSOUT=A
//SYSLMOD DD DSN=SYS1.NCPLOAD,DISP=SHR
//ANCPMOD1 DD DSN=SYS1.SNCPMOD1,DISP=SHR
//ABALMOD1 DD DSN=SYS1.SBALMOD1,DISP=SHR
/*
```

## NCP Input Produced by NDF (NPSI)

This section contains examples of NCP statements produced by the NPSI part of generation for input to VTAM. Also, examples are included of the default NPSI naming conventions and of the order of the statements generated for SVCs.

```

 OPTIONS USERGEN=X25NPSI,NEWDEFN=(YES,,SUPP)
BAKLOCAL PCCU DUMPDS=DUMPX25,SUBAREA=1
OTHLOCAL PCCU DUMPDS=DUMPX25,SUBAREA=16
REMTAM PCCU DUMPDS=DUMPX25,SUBAREA=2
*
BAKBUILD BUILD NPA=YES,VERSION=V5R2,LOADLIB=LNCP11,TYPGEN=NCP,MODEL=374*
 5,USGTIER=3,X25.USGTIER=3,NETID=NETID1,MAXSUBA=63,SUBARE*
 A=4,BFRS=124,LTRACE=4,OLT=YES,DR3270=NO,BRANCH=8000,TRAC*
 E=(YES,100),NEWNAME=SA324GN,X25.PREFIX=S,X25.IDNUMH=02,T*
 YPSYS=OS,X25.SNAP=YES,X25.MCHCNT=6,X25.MAXPIU=64K
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD,RIMM,NAKLIM*
 ,SESSION,SSPAUSE,XMTLMT)
BAKHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=1,DELAY=0*
 .1,TIMEOUT=30
OTHHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16,DELAY=*
 0.1,TIMEOUT=30
REMHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=2,DELAY=0*
 .1,TIMEOUT=30
PATH14 PATH DESTSA=(1),ER0=(1,1),ER1=(1,1),ER2=(1,1)
PATH116 PATH DESTSA=(16),ER0=(16,1),ER1=(16,1),ER2=(16,1)
PATH42 PATH DESTSA=(5,2,3),ER0=(5,2),ER1=(5,3),ER2=(3,4)
 LUDRPOOL NUMTYP1=10,NUMTYP2=10,NUMILU=2

SDLC6 GROUP LNCTL=SDLC,TYPE=NCP,DIAL=YES,REPLYTO=20

SL6 LINE ADDRESS=6,CLOCKNG=EXT,NEWSYNC=NO,DUPLEX=HALF,NRZI=NO,CALL*
 =INOUT,RETRIES=(7,0,0)
SP6 PU ISTATUS=INACTIVE,PUTYPE=2

SDLC5 GROUP LNCTL=SDLC,TYPE=NCP,DIAL=NO,REPLYTO=3,DISCNT=NO,ISTATUS=*
 INACTIVE

SL5 LINE ADDRESS=5,CLOCKNG=EXT,NEWSYNC=NO,DUPLEX=HALF,NRZI=NO,NPAC*
 OLL=YES,RETRIES=(3)
 SERVICE ORDER=(SP5)
SP5 PU ADDR=C2,MAXDATA=261,MAXOUT=3,PUTYPE=2,NPACOLL=YES,VPACING=2*
 ,PACING=1
SU5 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,NPACOLL=YES

GRNPA GROUP LNCTL=SDLC,TYPE=NCP,DIAL=NO,DISCNT=NO,VIRTUAL=YES,NPARSC*
 =YES,ISTATUS=INACTIVE
LNNPA LINE LINEFVT=NPAVFVT
PUNPA PU PUTYPE=2,PUFVT=NPAPFVT
LUNPA LU MAXCOLL=7,LUFVT=(NPALFVT,NPALFVT),VPACING=0,LOCADDR=1

***** X.25 NPSI INPUT
* ALLOW 3 VCCPT AND OUFT ENTRIES
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES
*
 X25.VCCPT INDEX=1,MAXPKTL=128,VWINDOW=1,INSLW=(100,50)
*

```

## Example A—NPSI Basic Functions

```
X25.VCCPT INDEX=2,MAXPKTL=128,VWINDOW=2,INSLOW=(100,50)
*
X25.VCCPT INDEX=3,MAXPKTL=128,VWINDOW=3,INSLOW=(100,50)
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2,OPTFACL=420707430202
*
X25.OUFT INDEX=3,OPTFACL=03430102420707430303,USRFIELD=12345678*
90

*

*
MCH SL7: NO DATE/GATE BUT SVC INN

X25.MCH ADDRESS=7,ANS=CONT,LCGDEF=(0,7),FRMLGTH=131,MWINDOW=7,*
STATION=DTE,TPTIMER=3,TDTIMER=1,NPRETRY=7,NDRETRY=1,LLCL*
IST=(LLC0,LLC2,LLC3,LLC5),LCN0=USED,SPAN=X2501,DBIT=NO,P*
AD=INTEG,TRAN=EVEN,PWPROT=YES,ITRACE=NO,SPEED=9600,T1TIM*
ER=1,SVCINN=2,SHM=YES,SDRTCNT=3,SDRTIME=10
*
X25.LCG LCGN=0
*
*
* PVC FOR INTEGRATED PAD FOR USE WITH TSO AND WITH PASSWORD PROTECTION
*
X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC5,SPAN=X2501,MAXDATA=2000*
,SSCPFM=USSNTO,MODETAB=MODTWX,DLOGMOD=TWXCONT,TERM=TWX,U*
SSTAB=USSTWX
*
*** 2 LINE'S FOR SUBAREA DIAL(2 SVC'S RESERVED FOR SVC INN)
*
X25.LINE DSTNODE=INN,TYPE=S
X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
X25.LINE DSTNODE=INN,TYPE=S
X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
*
* DEFINE 7 SVC TO BE SHARED BETWEEN INN SVC'S(MAX OF 2 AT A TIME)
* AND BNN SVC (MAX OF 5 AT A TIME)
*
X25.VC LCN=(1,7),VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1,SPAN=X2*
501
*

*
MCH SL0: NO DATE/GATE BUT TRANSPARENT PAD

X25.MCH ADDRESS=0,ANS=CONT,LCGDEF=(0,4),FRMLGTH=131,MWINDOW=2,*
STATION=DTE,XMONLNK=YES,TPTIMER=3,TDTIMER=1,NPRETRY=7,ND*
RETRY=1,LLCLIST=(LLC0,LLC2,LLC3,LLC5),LCN0=USED,SPAN=X25*
01,DBIT=NO,PAD=TRANSP,TRAN=EVEN,ITRACE=YES,SPEED=9600,T1*
TIMER=1
*
X25.LCG LCGN=0
*
*** X25 LINE FOR PSH (PVC) LINE NAME=SL0000 PU=SP0000 LU=SU0000
*
```

## Example A—NPSI Basic Functions

```

X25.LINE LCN=0,VCCINDX=3,LLC=LLC2,TYPE=P,SPAN=X2501
*
X25.PU PUTYPE=2,ADDR=01,MAXDATA=265,VPACING=2,PACING=2,ISTATUS*
 =INACTIVE
X25.LU LOCADDR=2,USSTAB=USSTD3,MODETAB=MODTD3
*
* 4 SVC'S: 1 IN, 2 IN/OUT, 1 OUT
*
X25.VC LCN=1,VCCINDX=2,TYPE=S,CALL=IN,OUFINDX=2,SPAN=X2501
X25.VC LCN=(2,3),VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=3,SPAN=X2*
 501
X25.VC LCN=4,VCCINDX=3,TYPE=S,CALL=OUT,OUFINDX=1,SPAN=X2501
*

* MCH SL1: GATE WITH SUBADDRESSING & PU T2.1

X25.MCH ADDRESS=1,ANS=CONT,LCGDEF=((0,5),(1,0)),FRMLGTH=131,MW*
 INDOV=3,STATION=DTE,XMONLNK=YES,SHUTD=NOINVCLR,TPTIMER=2*
 ,TDTIMER=3,NPRETRY=3,NDRETRY=3,LCN0=USED,PAD=INTEG,TRAN=*
 EVEN,GATE=GENERAL,SUBADDR=YES,LLCLIST=(LLC0,LLC2,LLC3,LL*
 C4,LLC5),LLC0=(0,2),LLC2=8,LLC3=9,LLC4=(1,4,3,7),LLC5=(6*
 ,5),SPAN=X2501,DBIT=NO
*
*
X25.LCG LCGN=0
*
* PVC FOR PCNE: LINK NAME=SL1000 PU=SP1000 LU=SU1000
*
X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC0,SPAN=X2501,MAXDATA=2000
*
***** X25 LINE FOR INN ***THIS SIDE IS SECONDARY****
*
SL1001 X25.LINE LCN=1,VCCINDX=3,MONLINK=YES,DSTNODE=INN,LLC=LLC3,SPAN*
 =X2501,TYPE=P
SP1001 X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE,SPAN=2*
 501
*
***** X25 LINE FOR INN ****THIS SIDE IS PRIMARY****
*
SL1002 X25.LINE LCN=2,VCCINDX=3,MONLINK=YES,DSTNODE=INN,LLC=LLC3,SPAN*
 =X2501,TYPE=P
SP1002 X25.PU PUTYPE=4,TGN=4,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE,SPAN=2*
 501
*
***** X25 LINE FOR BNN QLLC *****
*
SL1003 X25.LINE LCN=3,VCCINDX=3,LLC=LLC3,SPAN=X2501,TYPE=P
SP1003 X25.PU PUTYPE=2,ADDR=01,MAXDATA=265,VPACING=2,PACING=1,SPAN=X2*
 501,ISTATUS=INACTIVE
SU1003 X25.LU LOCADDR=2,USSTAB=USSTD3,MODETAB=MODTD3
*
* *** PVC 4 FOR PU2.1 ON BNN QLLC ***
* *** WITH DEPENDENT LU'S ***
*
X25.LINE LCN=4,VCCINDX=3,LLC=LLC3,SPAN=X2501,TYPE=P
*
X25.PU PUTYPE=2,ADDR=01,VPACING=2,XID=YES,DATMODE=HALF,PACING=*

```

## Example A—NPSI Basic Functions

```

1,SPAN=X2501,ISTATUS=INACTIVE,ANS=CONT
*
X25.LU LOCADDR=2
*

*
*
*
*** PVC 5 FOR PU2.1 ON BNN QLLC ***
*** WITH INDEPENDENT LU'S ***
*
X25.LINE LCN=5,VCCINDX=3,LLC=LLC3,SPAN=X2501,TYPE=P
*
X25.PU PUTYPE=2,ADDR=01,VPACING=2,XID=YES,DATMODE=HALF,PACING=*
1,SPAN=X2501,ISTATUS=INACTIVE,ANS=CONT
*
X25.LU LOCADDR=0,MODETAB=LENMODE
*
*
** VC USING SUBADDRESSING TO SELECT THE LLC TYPE *****
*
X25.LCG LCGN=1
**
X25.VC LCN=0,VCCINDX=3,OUFINDX=1,CALL=IN,SPAN=X2501,TYPE=SWITC*
HED

* MCH SL2 : GATE WITH USER DEFINED RELATION BETWEEN
* FIRST CUD BYTE AND CTCP/LLC

X25.MCH ADDRESS=2,ANS=CONT,LCGDEF=(0,2),FRMLGTH=131,MWINDOW=3,*
STATION=DTE,TPTIMER=3.1,TDTIMER=2,NPRETRY=7,NDRETRY=1,LC*
NO=USED,PAD=INTEG,TRAN=EVEN,GATE=GENERAL,SPAN=X2501,LLCL*
IST=(LLC0,LLC2,LLC3,LLC4,LLC5),CUD0=(00,02,21,22,NULL,03*
,B0,C1,C5,CC,FF),CTCP=(00,80,01,02,85,85,85,80,85,80,85)

*
* 3 CTCP'S FOR GATE.CUD0 SELECTIONS: CTCP0=00,CTCP1=21,CTCP2=22
* PCNE WILL BE SELECTED BY CUD0=02,C1,CC
* PAD WILL BE SELECTED BY CUD0=NULL,03,B0,C5,FF
*
* FOR CALLOUT IF L=7(PAD) THE CALL OUT PKT WILL HAVE A NULL CUD
* FOR CALLOUT IF L=. (PCNE) THE CALL OUT PKT WILL HAVE 02 IN CUD0

X25.LCG LCGN=0
*
*** PVC FOR PCNE *****
*
X25.VC LCN=0,VCCINDX=2,LLC=LLC0,MAXDATA=2000,TYPE=P
*
** 2 SVC'S *****
*
X25.VC LCN=(1,2),VCCINDX=3,OUFINDX=1,CALL=INOUT,TYPE=SWITCHED

* MCH SL3: DATE MCH

*
X25.MCH ADDRESS=3,ANS=CONT,GATE=DEDICAT,LCGDEF=(0,2),FRMLGTH=1*
31,MWINDOW=7,STATION=DTE,TPTIMER=3,TDTIMER=1,NPRETRY=7,N*
DRETRY=1,LLCLIST=(LLC0,LLC2,LLC3,LLC5),LCN0=NOTUSED,DBIT*

```

## Example A—NPSI Basic Functions

```

 =NO,SPEED=9600,T1TIMER=1,PAD=TRANSP,TRAN=EVEN
*
 X25.LCG LCGN=0

*
 X25.VC LCN=(1,2),VCCINDX=2,TYPE=S,CALL=INOUT,OUFINDX=2
*

* MCH XL11: 7 VC'S FOR FAST CONNECT WITH 2 CTCP'S

*

XL12 X25.MCH ADDRESS=4,PUNAME=XP12,LUNAME=(XU12,XU22),ANS=CONT,LCGD*
 EF=(0,7),FRMLGTH=131,MWINDOW=7,STATION=DTE,TPTIMER=3,TDT*
 IMER=1,SPAN=X2501,NPRETRY=7,NDRETRY=3,GATE=GENERAL,LLCLI*
 ST=LLC4,LCN0=NOTUSED,ITRACE=NO,LOGAPPL=(GBGTPLS,GBGTPLT)*
 ,CONNECT=SUBD,SUBD=(0,9,1),CTCP=(0,1,1)
*
 X25.LCG LCGN=0
*
 X25.VC LCN=(1,7),VCCINDX=3,TYPE=S,CALL=INOUT
*
* DEFINE THE MAX NB OF LU'S USABLE AT THE SAME TIME ON CTCP0
*
 X25.FCG QTY=(5),CTCPNO=(0),SPAN=X2501,MAXDATA=2500,PRFLINE=XL1*
 2,PRFPU=XP12,PRFLU=XU21,SUFFIX=0001
*
* DEFINE THE MAX NB OF LU'S USABLE AT THE SAME TIME ON CTCP1
*
 X25.FCG QTY=(5),CTCPNO=(1),SPAN=X2501,MAXDATA=2500,PRFLINE=XL2*
 2,PRFPU=XP22,PRFLU=XU22,SUFFIX=0001
*

 X25.END
S25P7A GROUP DIAL=NO,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINEA*
 DD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LNC*
 TL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),T*
 YPE=NCP,USERID=(5688035,BALPBDT,NORECMS),XIO=(BALNAVXL,B*
 ALNAVXS,BALNAVXI,BALNAVXK)
SL7000 LINE IPL=NO,SPAN=X2501,UACB=XA7000
SC7000 SERVICE
SP7000 PU ADDR=01,PACING=1,PUDR=NO,PUTYPE=1,XID=NO,SPAN=X2501,MAXDATA*
 =2000
SU7000 LU LOCADDR=0,SPAN=X2501,TERM=TWX,DLOGMOD=TWXCONT,SSCPFM=USSNTO*
 ,USSTAB=USSTWX,MODETAB=MODTWX,VPACING=(2,1),LUDR=NO
S25P0A GROUP DIAL=NO,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINEA*
 DD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LNC*
 TL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),T*
 YPE=NCP,USERID=(5688035,BALPBDT,NORECMS),XIO=(BALNAVXL,B*
 ALNAVXS,BALNAVXI,BALNAVXK)
SL0000 LINE SPAN=X2501,IPL=NO,UACB=XA0000
SC0000 SERVICE
SP0000 PU ADDR=01,MAXDATA=265,PACING=2,ISTATUS=INACTIVE,VPACING=2,PUT*
 YPE=2,XID=NO,PUDR=NO
SU0000 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,LUDR=NO
S25P1A GROUP DIAL=NO,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINEA*
 DD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LNC*
 TL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),T*

```

## Example A—NPSI Basic Functions

```

 YPE=NCP,USERID=(5688035,BALPBDT,NORECMS),XIO=(BALNAVXL,B*
 ALNAVXS,BALNAVXI,BALNAVXK)
SL1000 LINE IPL=NO,SPAN=X2501,UACB=XA1000
SC1000 SERVICE
SP1000 PU ADDR=01,PACING=1,PUDR=NO,PUTYPE=1,XID=NO,SPAN=X2501,MAXDATA*
 =2000
SU1000 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
SL1001 LINE SPAN=X2501,IPL=NO,MONLINK=YES,UACB=XA1001
SC1001 SERVICE
SP1001 PU ANS=CONT,TGN=2,ISTATUS=INACTIVE,SPAN=2501,PUTYPE=4,PUDR=NO
SL1002 LINE SPAN=X2501,IPL=NO,MONLINK=YES,UACB=XA1002
SC1002 SERVICE
SP1002 PU ANS=CONT,TGN=4,ISTATUS=INACTIVE,SPAN=2501,PUTYPE=4,PUDR=NO
SL1003 LINE SPAN=X2501,IPL=NO,UACB=XA1003
SC1003 SERVICE
SP1003 PU ADDR=01,MAXDATA=265,PACING=1,ISTATUS=INACTIVE,SPAN=X2501,VP*
 ACING=2,PUTYPE=2,PUDR=NO
SU1003 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,LUDR=NO
SL1004 LINE SPAN=X2501,IPL=NO,UACB=XA1004
SC1004 SERVICE
SP1004 PU ADDR=01,ANS=CONT,DATMODE=HALF,PACING=1,ISTATUS=INACTIVE,SPA*
 N=X2501,VPACING=2,PUTYPE=2,XID=YES,PUDR=NO
SU1004 LU LOCADDR=2,LUDR=NO
SL1005 LINE SPAN=X2501,IPL=NO,UACB=XA1005
SC1005 SERVICE
SP1005 PU ADDR=01,ANS=CONT,DATMODE=HALF,PACING=1,ISTATUS=INACTIVE,SPA*
 N=X2501,VPACING=2,PUTYPE=2,XID=YES,PUDR=NO
SU1005 LU LOCADDR=0,MODETAB=LENMODE,LUDR=NO
S25P2A GROUP DIAL=NO,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINEA*
 DD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LNC*
 TL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),T*
 YPE=NCP,USERID=(5688035,BALPBDT,NORECMS),XIO=(BALNAVXL,B*
 ALNAVXS,BALNAVXI,BALNAVXK)
SL2000 LINE IPL=NO,UACB=XA2000
SC2000 SERVICE
SP2000 PU ADDR=01,PACING=1,PUDR=NO,PUTYPE=1,XID=NO,MAXDATA=2000
SU2000 LU LOCADDR=0,VPACING=(2,1),LUDR=NO
S25PG50B GROUP DIAL=NO,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINEA*
 DD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LNC*
 TL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),T*
 YPE=NCP,USERID=(5688035,BALFBDT,NORECMS),LOGAPPL=GBGTPLS*
 ,XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
XL120001 LINE IPL=NO,SPAN=X2501,UACB=XAG50000
SCG50000 SERVICE
XP120001 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU210001 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL120002 LINE IPL=NO,SPAN=X2501,UACB=XAG50001
SCG50001 SERVICE
XP120002 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU210002 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL120003 LINE IPL=NO,SPAN=X2501,UACB=XAG50002
SCG50002 SERVICE
XP120003 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU210003 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL120004 LINE IPL=NO,SPAN=X2501,UACB=XAG50003
```



## Example A—NPSI Basic Functions

```
SCG50003 SERVICE
XP120004 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU210004 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL120005 LINE IPL=NO,SPAN=X2501,UACB=XAG50004
SCG50004 SERVICE
XP120005 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU210005 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
S25PG51C GROUP DIAL=NO,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
 DD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LNC*
 TL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),T*
 YPE=NCP,USERID=(5688035,BALFBDT,NORECMS),LOGAPPL=GBGTPLT*
 ,XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
XL220001 LINE IPL=NO,SPAN=X2501,UACB=XAG51000
SCG51000 SERVICE
XP220001 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU220001 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL220002 LINE IPL=NO,SPAN=X2501,UACB=XAG51001
SCG51001 SERVICE
XP220002 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU220002 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL220003 LINE IPL=NO,SPAN=X2501,UACB=XAG51002
SCG51002 SERVICE
XP220003 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU220003 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL220004 LINE IPL=NO,SPAN=X2501,UACB=XAG51003
SCG51003 SERVICE
XP220004 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU220004 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
XL220005 LINE IPL=NO,SPAN=X2501,UACB=XAG51004
SCG51004 SERVICE
XP220005 PU ADDR=01,MAXDATA=2500,XID=NO,PACING=1,PUDR=NO,SPAN=X2501,PUT*
 YPE=1
XU220005 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),LUDR=NO
S25S3A GROUP DIAL=YES,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
 ADD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LI*
 NEAUT=YES,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATS*
 T,BALNATLS),TYPE=NCP,USERID=(5688035,BALSBDT,NORECMS),XI*
 O=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
SL3002 LINE CALL=IN,UACB=XA3002
SP3002 PU PUTYPE=(1,2)
SL3001 LINE CALL=IN,UACB=XA3001
SP3001 PU PUTYPE=(1,2)
S25S2B GROUP DIAL=YES,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
 ADD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSPW=NO,COMPACB=NO,LI*
 NEAUT=YES,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATS*
 T,BALNATLS),TYPE=NCP,USERID=(5688035,BALSBDT,NORECMS),XI*
 O=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
SL2002 LINE CALL=INOUT,UACB=XA2002
SP2002 PU PUTYPE=(1,2)
SL2001 LINE CALL=INOUT,UACB=XA2001
SP2001 PU PUTYPE=(1,2)
S25S1B GROUP DIAL=YES,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
```

## Example A—NPSI Basic Functions

```

 ADD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSWP=NO,COMPACB=NO,LI*
 NEAUT=YES,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATS*
 T,BALNATLS),TYPE=NCP,USERID=(5688035,BALSBDT,NORECMS),XI*
 O=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
SL1100 LINE CALL=IN,SPAN=X2501,UACB=XA1100
SP1100 PU SPAN=X2501,PUTYPE=(1,2)
S25S0B GROUP DIAL=YES,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
 ADD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSWP=NO,COMPACB=NO,LI*
 NEAUT=YES,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATS*
 T,BALNATLS),TYPE=NCP,USERID=(5688035,BALSBDT,NORECMS),XI*
 O=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
SL0004 LINE CALL=OUT,SPAN=X2501,UACB=XA0004
SP0004 PU SPAN=X2501,PUTYPE=(1,2)
SL0003 LINE CALL=INOUT,SPAN=X2501,UACB=XA0003
SP0003 PU SPAN=X2501,PUTYPE=(1,2)
SL0002 LINE CALL=INOUT,SPAN=X2501,UACB=XA0002
SP0002 PU SPAN=X2501,PUTYPE=(1,2)
SL0001 LINE CALL=IN,SPAN=X2501,UACB=XA0001
SP0001 PU SPAN=X2501,PUTYPE=(1,2)
S25S7C GROUP DIAL=YES,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
 ADD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSWP=NO,COMPACB=NO,LI*
 NEAUT=YES,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATS*
 T,BALNATLS),TYPE=NCP,USERID=(5688035,BALSBDT,NORECMS),XI*
 O=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
SL7007 LINE CALL=INOUT,SPAN=X2501,UACB=XA7007
SP7007 PU SPAN=X2501,PUTYPE=(1,2)
SL7006 LINE CALL=INOUT,SPAN=X2501,UACB=XA7006
SP7006 PU SPAN=X2501,PUTYPE=(1,2)
SL7005 LINE CALL=INOUT,SPAN=X2501,UACB=XA7005
SP7005 PU SPAN=X2501,PUTYPE=(1,2)
SL7004 LINE CALL=INOUT,SPAN=X2501,UACB=XA7004
SP7004 PU SPAN=X2501,PUTYPE=(1,2)
SL7003 LINE CALL=INOUT,SPAN=X2501,UACB=XA7003
SP7003 PU SPAN=X2501,PUTYPE=(1,2)
S25S7B GROUP DIAL=YES,LEVEL2=BALNAVL2,LEVEL3=BALNAVL3,LEVEL5=NCP,LINE*
 ADD=NONE,COMPTAD=NO,COMPOWN=YES,COMPSWP=NO,COMPACB=NO,LN*
 CTL=SDLC,NPACOLL=NO,TIMER=(BALNATER,,BALNATST,BALNATLS),*
 TYPE=NCP,USERID=(5688035,BALIBDT,NORECMS),PUTYPE=4,SHM=Y*
 ES,ACTIVTO=0.0,BRKCON=NONE,XIO=(BALNAVXL,BALNAVXS,BALNAV*
 XI,BALNAVXK)
SL7GGI LINE CALL=INOUT,UACB=XA7GGI
SP7GGI PU ISTATUS=INACTIVE,PUTYPE=4
SL7GGH LINE CALL=INOUT,UACB=XA7GGH
SP7GGH PU ISTATUS=INACTIVE,PUTYPE=4
SNET11 GROUP DIAL=NO,BERPROC=BALNUMBER,COMPACB=YES,COMPTAD=YES,COMPOWN*
 =YES,COMPSWP=YES,LEVEL2=BALNAML2,LEVEL3=BALNAML3,LEVEL5=*
 NCP,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALLAP4,,BALLAP4,BALLAP*
 4),TYPE=NCP,USERID=(5688035,BALMBDT,NORECMS,P),XIO=(BALN*
 AMXL,BALNAMXS,BALNAMXI,BALNAMXK)
SL7 LINE ADDRESS=(7,FULL),SPAN=X2501,XMONLNK=NO,UACB=(X25A7X,X25A7*
 R)
SC7 SERVICE
SP7 PU ADDR=01,ANS=CONT,SPAN=X2501,MAXDATA=261,XID=NO,PUTYPE=1
SU7 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=INACTIVE
SNET12 GROUP DIAL=NO,BERPROC=BALNUMBER,COMPACB=YES,COMPTAD=YES,COMPOWN*
 =YES,COMPSWP=YES,LEVEL2=BALNAML2,LEVEL3=BALNAML3,LEVEL5=*
 NCP,LNCTL=SDLC,NPACOLL=NO,TIMER=(BALLAP4,,BALLAP4,BALLAP*
 4),TYPE=NCP,USERID=(5688035,BALNBDT,NORECMS,P),XIO=(BALN*
```

## Example A—NPSI Basic Functions

```
 AMXL,BALNAMXS,BALNAMXI,BALNAMXK)
SL0 LINE ADDRESS=(0,FULL),SPAN=X2501,XMONLNK=YES,UACB=(X25A0X,X25A*
 0R)
SC0 SERVICE
SP0 PU ADDR=01,ANS=CONT,SPAN=X2501,MAXDATA=261,XID=NO,PUTYPE=1
SU0 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=INACTIVE
SL1 LINE ADDRESS=(1,FULL),SPAN=X2501,XMONLNK=YES,UACB=(X25A1X,X25A*
 1R)
SC1 SERVICE
SP1 PU ADDR=01,ANS=CONT,SPAN=X2501,MAXDATA=261,XID=NO,PUTYPE=1
SU1 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
SL2 LINE ADDRESS=(2,FULL),SPAN=X2501,XMONLNK=NO,UACB=(X25A2X,X25A2*
 R)
SC2 SERVICE
SP2 PU ADDR=01,ANS=CONT,SPAN=X2501,MAXDATA=261,XID=NO,PUTYPE=1
SU2 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
SU2L1 LU LOCADDR=1,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
SU2L2 LU LOCADDR=2,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
SL3 LINE ADDRESS=(3,FULL),XMONLNK=NO,UACB=(X25A3X,X25A3R)
SC3 SERVICE
SP3 PU ADDR=01,ANS=CONT,MAXDATA=261,XID=NO,PUTYPE=1
SU3 LU LOCADDR=0,VPACING=(2,1),ISTATUS=ACTIVE
XL12 LINE ADDRESS=(4,FULL),SPAN=X2501,XMONLNK=NO,UACB=(X25A4X,X25A4*
 R)
SC4 SERVICE
XP12 PU ADDR=01,ANS=CONT,SPAN=X2501,MAXDATA=261,XID=NO,PUTYPE=1
XU12 LU LOCADDR=0,LOGAPPL=GBGTPLS,SPAN=X2501,VPACING=(2,1),ISTATUS=*
 ACTIVE
XU22 LU LOCADDR=1,LOGAPPL=GBGTPLT,SPAN=X2501,VPACING=(2,1),ISTATUS=*
 ACTIVE

CAGROUP GROUP LNCTL=CA,CA=TYPE6,DELAY=0.0,NCPCA=ACTIVE
CALINE1 LINE ADDRESS=08
CA1PU PU PUTYPE=5
 GENEND INIT=BALINIMD,LINOPT=BSC,TMRTICK=BALTICK,UGLOBAL=BALNMG*
 OP

```

## VTAM Main Switched Major Node

This example illustrates some of the IDNUM usages that go with the NPSI generation.

```

* SWITCHED MAJOR NODE TO BE USED IN CORRELATION WITH SA324GN GEN

**
SWS25 VBUILD MAXGRP=20, *
 MAXNO=20, *
 TYPE=SWNET
**

* ENTRY FOR PCNE THAT RECEIVES DATA BEFORE SDT *

*
PCNEPBK PU ADDR=01, PCNE FOR TEST WITH DXPCNEBK *
 IDBLK=003, AND PCNEBK APPL. DATA IS *
 IDNUM=22225, QUEUED IN NPSI UNTIL SDT, BE THE*
 DISCNT=NO,MAXOUT=6, SDT AFTER ACTLU OR UNBIND W/O *
 MAXDATA=1024, ACTLU. NOTICE THE DISCNT=NO... *
 PACING=1, *
 VPACING=2, *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNELBK LU LOCADDR=0,LOGAPPL=TD5SR MAXOUT=6 GOES W LOGAPPL
**

* ENTRIES FOR PCNE TO PCNE COMMUNICATION *
* (BACK TO BACK) *

*
PCNEPI PU ADDR=01, PCNE TO PCNE INBOUND *
 IDBLK=003, *
 IDNUM=22223, ZZZZ OF OUT CALLER +1 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXOUT=1, TO LET LOGON MSG IN AFTER ACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNELI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
**
PCNEPO PU ADDR=01, PCNE TO PCNE OUTBOUND *
 IDBLK=003, *
 IDNUM=22222, EQUAL ZZZZ *
 MAXPATH=6, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNEP1 PATH DIALNO=10604016500201*22222, ZZZZ=22222=IDNUM *

```

## Example A—NPSI Basic Functions

```

GRPNM=S25S0B,GID=2,PID=20 S25S0B=GROUP OF SWITCHED
* VC'S ON INVOLVED MCH IN GENERATION
PCNEP3 PATH DIALNO=10604016900201*22222,
* GRPNM=S25S0B,GID=2,PID=22
PCNELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*

* *
* ENTRY FOR INCOMING AND OUTGOING CALLS FOR PADX ON MCH0 *
* *

*

*
* DEFAULT IDNUM VALUES FOR INCOMING CALLS ARE BUILT AS FOLLOW:
* IDNUM=02012
* ..012 FOR 9TH SVC FROM LAST SVC (9*2)=12(HEX)
* 02... COME FROM X25.IDNUMH IN NPSI BUILD MACRO

*
PADXPO PU ADDR=01, ENTRY USED FOR IN AND OUTCALL
* IDBLK=003,
* IDNUM=02012, FOR 9TH SVC FROM END OF GEN(9*2)=12
* MAXPATH=6,
* DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU
* MAXDATA=1024, SEGMENTING NOT SUPPORTED
* PACING=1, PACING=1 IS REQUIRED
* VPACING=2,
* ANS=CONT, FOR SW SESSION CONTINUATION
* PUTYPE=1,
* SSCPFM=USSSCS
PADP1 PATH DIALNO=10605023650101*02012, ZZZZ=02012=IDNUM
* GRPNM=S25S0B,GID=2,PID=50
PADXLO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*

* *
* ENTRY FOR INCOMING AND OUTGOING CALLS FOR BNN QLLC/PSH *
* *

*
PSHPI PU ADDR=01,
* IDBLK=017,
* IDNUM=41264, IDNUM OF REMOTE PU
* MAXPATH=6,
* DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU
* MAXDATA=265, SEGMENTING SUPPORTED
* PACING=3,
* ANS=CONT, FOR SW SESSION CONTINUATION
* VPACING=3,
* PUTYPE=2,
* SSCPFM=USSSCS
PSHP1 PATH DIALNO=10604016420201, L=2 FOR PSH
* GRPNM=S25S0B,GID=2,PID=30 S25S0B=GROUP OF SWITCHED
* VC'S ON INVOLVED MCH IN GENERATION
PSHP2 PATH DIALNO=10604016430201, L=3 FOR BNN QLLC
* GRPNM=S25S0B,GID=2,PID=31
PSHLI LU LOCADDR=2,MODETAB=MODTD3

```

## Example A—NPSI Basic Functions

```

*

*
* ENTRY FOR SUBADDRESSING ON MCH1 FOR NON SNA VIRTUAL CIRCUITS *
*

*
SAPI PU ADDR=01, GATE OR PCNE OR PAD CALL IN *
 IDBLK=003,
 IDNUM=0200A, FOR 5TH SVC FROM LAST SVC (5*2)=A *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2,
 PUTYPE=1,MAXOUT=6, TO QUEUE DATA UNTIL SDT *
 SSCPFM=USSSCS
SALI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*

*
* ENTRIES FOR GATE AND PADI *
* FOR GATE, CALLIN IS ALWAYS SIMULATED TO VTAM. PATH NOT NEEDED *
*

*
GATEPI PU ADDR=01, PADI INCALL OR 1ST GATE VC USED *
 IDBLK=003, (IN OR OUT CALL) *
 IDNUM=02008, FOR 4TH SVC FROM LAST SVC (4*2)=8 *
 MAXOUT=6, TO QUEUE INBOUND MSG COMING BEFORE SDT *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2,
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1,
 SSCPFM=USSSCS
PADILI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*
**
*
GATEPO PU ADDR=01, 2ND GATE VC USED(IN OR OUT CALL) *
 IDBLK=003,
 IDNUM=02006, FOR 3RD SVC FROM LAST SVC (3*2)=6 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2,
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1,
 SSCPFM=USSSCS
GATELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*

*
* ENTRIES FOR USER DEFINED CUD0 (CALLOUT) *
*

```

## Example A—NPSI Basic Functions

```

*
UDCUDP1 PU ADDR=01, TO BE USED FOR PCNE *
 IDBLK=003, *
 MAXPATH=4, *
 IDNUM=55544, CUD123 FOR CALLIN,EQ ZZZZ FOR CALL OUT *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 PUTYPE=1, *
 SSCPFM=USSSCS USSTAB=USSTABZ
UDCUDH1 PATH DIALNO=106050236.0101*55544, .MEANS USER DEFINED CUDO *
 GRPNM=S25S2B,GID=2,PID=96 AND NOTHING IN CUD1 2 3
UDCUDU1 LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*
UDCUDP2 PU ADDR=01, TO BE USED FOR PCNE *
 IDBLK=003, *
 MAXPATH=4, *
 IDNUM=55555, CUD123 FOR CALLIN, EQ ZZZZ FOR CALLOUT *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 PUTYPE=1, *
 SSCPFM=USSSCS USSTAB=USSTABZ
UDCUDH2 PATH DIALNO=106050236:0101*55555, MEANS USER DEFINED CUDO *
 GRPNM=S25S2B,GID=2,PID=97 AND ZZZZ+1 IN CUD1,2,3.
UDCUDU2 LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*

*
* ENTRIES FOR OUTGOING CALLS FOR PADI ON MCH2 *
*

*

*
* DEFAULT IDNUM-1 VALUES FOR OUTCALLS ARE NO LONGER USED
* ZZZZ MUST BE SPECIFIED IN THE DIALNO

*
PADIPO PU ADDR=01, *
 IDBLK=003, *
 IDNUM=02005, EQUAL ZZZZ BELOW(ODD NO TO AVOID *
 MAXPATH=6, INTERFERENCE W CALL IN) *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PADP2 PATH DIALNO=10605023650201*02005, L=5(PAD) *
 GRPNM=S25S2B,GID=2,PID=60
PADILO LU LOCADDR=0,TERM=TWX REQUIRED FOR NON SNA X25 DEVICES
*

*

```

## Example A—NPSI Basic Functions

```
* ENTRIES FOR DATE VC'S, TO BE USED FOR PCNE OR PADX *
* FOR DATE, CALLIN IS ALWAYS SIMULATED TO VTAM. PATH NOT NEEDED *

*
DATEPI PU ADDR=01, 1ST DATE VC USED (IN OR OUT CALL) *
 IDBLK=003, *
 IDNUM=02004, FOR 2ND SVC FROM LAST SVC (2*2)=4 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS *
DATELI LU LOCADDR=0
*
DATEPO PU ADDR=01, 2ND DATE VC USED (IN OR OUT CALL) *
 IDBLK=003, *
 IDNUM=02002, FOR 1ST SVC FROM LAST SVC (1*2)=2 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS *
DATELO LU LOCADDR=0
*
```



## VTAM Switched Major Node for SNA Type 2.1 and SVCSC

This section contains examples of SMN entries to be used for SNA type 2.1 and SVCSC.

```

*
* SWITCHED MAJOR NODE TO BE USED TO TEST NPSI SWITCHED SESSION CONT.
* AND LEN FUNCTIONS.
* NCP/NPSI GENERATION: SA324GN
*

*
SWLENSAD VBUILD MAXGRP=20,
 MAXNO=20,
 TYPE=SWNET
*
*

*
* THE FOLLOWING SWITH MAJOR NODE IS USED:
*
* - MP1006 IS A PU2.1 WITH LU6.2
*
* - MP2004 IS A PU2.1 WITH LU2
*
* - MP2006 IS A PU2.1 WITH LU6.2
*

*
SMP0006 PU ADDR=01,
 CPNAME=NODE16, EQUIVALENT TO IDBLK/IDNUM
 MAXPATH=1,
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU
 PACING=1,
 VPACING=2,
 PUTYPE=2,
 ANS=CONT,
 SSCPFM=USSSCS
*
LENP2 PATH DIALNO=10604016430201, COMMUNICATES WITH SMP4006 VIA
 GRPNM=S25S0B, SAME MCH(APPL NEEDED IF CALLOUT
 GID=2, IS DONE)
 PID=47
*
SMU0006 LU LOCADDR=0,
 MODETAB=LENMODE
*

*
SMP4004 PU ADDR=01,
 CPNAME=NODE24, REPLACES IDNUM/IDBLK FOR PUT2.1
 PUTYPE=2,
 ANS=CONT,
 VPACING=2,
 PACING=1,
 MAXPATH=1,
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU

```

## Example A—NPSI Basic Functions

```

 SSCPFM=USSSCS
*
LENP1 PATH DIALNO=10604016430201, L=3 FOR QLLC BNN *
 GRPNM=S25S0B, GENERATED GROUP OF SWITCHED *
 GID=2, VC'S VCS *
 PID=47
*
SMU4004 LU LOCADDR=1
*

*
SMP4006 PU ADDR=01, *
 CPNAME=NODE26, *
 MAXPATH=1, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 PACING=1, *
 VPACING=2, *
 PUTYPE=2, *
 ANS=CONT, *
 SSCPFM=USSSCS
*
LENP3 PATH DIALNO=10604016430201, COMMUNICATES WITH SMP0006 VIA *
 GRPNM=S25S0B, SAME MCH (APPL NEEDED IF CALLOUT*
 GID=2, TO BE DONE) *
 PID=47
*
SMU4006 LU LOCADDR=0, *
 MODETAB=LENMODE
*

*
* X25 SVC SUBAREA COMMUNICATION(SVCSC) *
*
* SWITCHED MAJOR NODE FOR X25 FOR BOTH NCP'S *
*
* SHM SUPPORTED ON BOTH SIDES, EITHER SIDE CAN CALL *
*
* CONFIG = VTAM1(SA16)- NCP SA4---'X25NW'----NCP SA5---VTAM2(SA2)*
*

*
**
** FOR OUTGOING CALL ISSUED FROM VTAM1 OR INCOMING CALL FROM VTAM2
SVCSCPU5 PU PUTYPE=4, *
 ADDR=05, SA 5 IS PRIMARY *
 TGN=ANY, *
 SUBAREA=5, SA NO OF OTHER SIDE *
 IDNUM=AAAAA, MUST BE SAME ON BOTH SIDES *
 NETID=NETID1, OTHER SIDE'S NETID (VTAM2) *
 ANS=CONT, *
 MAXDATA=1024, *
 MAXPATH=1
*
SVCSCPA5 PATH DIALNO=10604016410201,GID=51,PID=30, *L=1 FOR SVCSC *
 GRPNM=S25S7B, SVCSC LINES GROUP NAME *
 VERIFY=IN, VERIFY INCOMING CALL ADDR *
 VERID=106040164, CALLER ADDR *

```

## Example A—NPSI Basic Functions

```

SHM=YES,
SHMTIM=45 CLEAR AFTER 45 SEC OF INACTIVITY
*
*
** THIS PART IS IN VTAM2 SWITCH MAJ NODE BUT IS COPIED HERE FOR INFO
*
** FOR INCOMING CALL RECEIVED BY VTAM2 SIDE OR OUTCALL BY VTAM2
SVCSCPU4 PU PUTYPE=4,
 ADDR=04, SA 4 IS SECONDARY
 TGN=2, TG 2 ALSO USED IN PATH IN NCP
 SUBAREA=4, SA NO OF OTHER SIDE
 IDNUM=AAAAA, MUST BE SAME ON BOTH SIDES
 NETID=NETID1, VTAM1 NETID(SAME NW AS VTAM1)
 ANS=CONT,
 MAXDATA=1024,
 MAXPATH=1
*
SVCSCPA4 PATH DIALNO=10604016910201,GID=52,PID=30, *L=1
 GRPNM=C25S0A, SVCSC LINES GROUP NAME
 SHM=YES,
 SHMTIM=33 CLEAR AFTER 33 SEC(IGNORED(LOWEST))

*
* X25 SVC SUBAREA COMMUNICATION(SVCSC)
*
* SWITCHED MAJOR NODE FOR X25 FOR BOTH NCP'S
*
* SHM NOT SUPPORTED AT ALL. CALLING ALLOWED FROM SA3 ONLY
*
* CONFIG = VTAM1(SA16)- NCP SA4---'X25NW'----NCP SA3---VTAM2(SA2)*
*

*
** FOR OUTGOING CALL ISSUED FROM VTAM1 OR INCOMING CALL FROM VTAM2
SVCSCPU3 PU PUTYPE=4,
 ADDR=03, SA 3 IS SECONDARY
 TGN=ANY,
 SUBAREA=3, SA NO OF OTHER SIDE
 IDNUM=AAA3, MUST BE SAME ON BOTH SIDES
 NETID=NETID1, OTHER SIDE'S NETID (VTAM2)
 ANS=CONT,
 MAXDATA=1024,
 MAXPATH=1
*
SVCSCPA3 PATH CALL=IN,GID=31,PID=30, *TO SA5
 GRPNM=S25S7B, SVCSC LINES GROUP NAME
 VERIFY=NONE NO CALLER ID VERIFICATION
*
*
** THIS PART IS IN VTAM2 SWITCH MAJ NODE BUT IS COPIED HERE FOR INFO
*
** FOR INCOMING CALL RECEIVED BY VTAM2 SIDE OR OUTCALL BY VTAM2
SVCSCPUD PU PUTYPE=4,
 ADDR=04,
 TGN=4, TG 4 ALSO USED IN PATH IN NCP
 SUBAREA=4, SA NO OF OTHER SIDE
 IDNUM=AAA3, MUST BE SAME ON BOTH SIDES

```

## Example A—NPSI Basic Functions

```
NETID=NETID1, VTAM1 NETID *
ANS=CONT, *
MAXDATA=1024, *
MAXPATH=1
*
SVCSCPAD PATH DIALNO=10604016910201,GID=32,PID=30, *TO SA4 *
GRPNM=T25S0A, SVCSC GROUP NAME *
CALL=OUT ALLOW OUT CALL ONLY *
```

## MODETAB for Password Protection and LU 6.2

This section contains examples of mode tables for password protection and LU type 6.2.

```

MTTWX TITLE 'MODE TABLE ENTRIES FOR TWX 3101 TERMINAL THROUGH X.25'
 PRINT NOGEN
 EJECT
MODTXX MODETAB
TWXFLIP MODEENT LOGMODE=TWXFLIP, FOR PAD PASSWORD PROTECTION X
 FMPROF=X'03', OF NPSI V3R1 AND UP X
 TSPROF=X'03', X
 PRIPROT=X'B1', X
 SECPROT=X'A0', X
 COMPROT=X'3080', X
 RUSIZES=X'8585', X
 DCODE=X'80', FOR DEVICES WHICH SUPPORT INP/ENP X
 PSERVIC=X'010000000000000000000000'
*
 0 1 2 3 4 5 6 7 8 9 1011
TWXCONT MODEENT LOGMODE=TWXCONT, FOR PAD PASSWORD PROTECTION X
 FMPROF=X'03', OF NPSI V3R1 AND UP X
 TSPROF=X'03', X
 PRIPROT=X'B1', X
 SECPROT=X'A0', X
 COMPROT=X'3040', X
 RUSIZES=X'8585', X
 DCODE=X'80', FOR DEVICES WHICH SUPPORT INP/ENP X
 PSERVIC=X'010000000000000000000000'
*
 0 1 2 3 4 5 6 7 8 9 1011
TWXFLIPP MODEENT LOGMODE=TWXFLIPP, X
 FMPROF=X'03', X
 TSPROF=X'03', X
 PRIPROT=X'B1', X
 SECPROT=X'A0', X
 COMPROT=X'3080', X
 RUSIZES=X'8585', X
 DCODE=X'00', X
 PSERVIC=X'010000000000000000000000'
*
 0 1 2 3 4 5 6 7 8 9 1011
TWXCONTP MODEENT LOGMODE=TWXCONTP, X
 FMPROF=X'03', X
 TSPROF=X'03', X
 PRIPROT=X'B1', X
 SECPROT=X'A0', X
 COMPROT=X'3040', X
 RUSIZES=X'8585', X
 DCODE=X'00', X
 PSERVIC=X'010000000000000000000000'
*
 0 1 2 3 4 5 6 7 8 9 1011
 MODEEND

 TITLE 'MODE TABLE ENTRIES FOR LU 6.2 THROUGH X.25'
LENMODE MODETAB
LU62MODE MODEENT LOGMODE=LU62MODE, ENTRY NAME X
 COMPROT=X'51B1', COMMON PROTOCOL X
 ENCR=0, ENCRYPTION X

```

## Example A—NPSI Basic Functions

```
FMPROF=19, FM PROFILE 19 X
PRIPROT=X'B0', PRIMARY PROTOCOL X
PSERVIC=X'0602000000000000000002300', X
PSNDPAC=9, PRIMARY SEND PACING X
RUSIZES=X'C3C3', RU SIZES X
SECPROT=X'B0', SECONDARY PROTOCOLS X
SRCVPAC=5, SECONDARY RECEIVE PACING COUX X
SSNDPAC=5, SECONDARY SEND PACING COUNT X
TSPROF=7, TS PROFILE X
TYPE=0 NEGOCIABLE BIND

*
SNASVCMG MODEENT LOGMODE=SNASVCMG, ENTRY NAME X
COMPROT=X'50A5', COMMON PROTOCOL X
ENCR=0, ENCRYPTION X
FMPROF=19, FM PROFILE 19 X
PRIPROT=X'B0', PRIMARY PROTOCOL X
PSERVIC=X'0602000000000000000002380', X
PSNDPAC=35, PRIMARY SEND PACING X
RUSIZES=X'8585', RU SIZES X
SECPROT=X'B0', SECONDARY PROTOCOLS X
SRCVPAC=7, SECONDARY RECEIVE PACING COUX X
SSNDPAC=7, SECONDARY SEND PACING COUNT X
TSPROF=7, TS PROFILE X
TYPE=0 NEGOCIABLE BIND

*
MODEEND
END
```

## Example B—NPSI Default and User-Defined IDNUM and Remote Loading

This section consists of three examples:

- Example B1 shows the use of default IDNUM values, and the definition of a link to example B3 on which the NCP can be loaded.
- Example B2 shows a configuration similar to example B1. However, this example illustrates the use of user-defined IDNUM values.
- Example B3 is of a remote NCP that contains a link to example B1 upon which this NCP can be loaded.

### NCP and NPSI Generation Input—Example B1

Example B1 shows the use of default IDNUM values, and the definition of a link to example B3 on which the NCP can be loaded.

```

* GENERATION NAME: EX3346Z *
* *
* - MCH1 (64) PVC0 FOR PCNE *
* 2SVC FOR PCNE PAD GATE *
* - MCH2 (65) PVC0 FOR CASUAL CONNECT *
* 3SVC FOR PSH,BNNQLLC AND CASUAL CONNECT *
* PVC4 FOR CASUAL CONNECT *
* PVC5 FOR GATE *
* - MCH3 (66) PVC0 FOR INN QLLC REMOTE LOADING *
* SVC1 FOR INN QLLC REMOTE LOADING *
* - MCH4 (67) SVC1 FOR X21 ACCESS (GATE) *

* IDNUM ARE USER DEFINED AS IDNUMT IS SPECIFIED IN X25.VC *

*
**
**
*
 OPTIONS USERGEN=X25NPSI,NEWDEFN=(YES,ECHO)
*
ATTACHED PCCU DUMPDS=DUMPX25,SUBAREA=16
*
BAKBUILD BUILD NPA=YES, X
 VERSION=V5R3, X
 LOADLIB=LNCP10, X
 TYPGEN=NCP, X
 MODEL=3745, X
 NETID=FRIBM412, X
 MAXSUBA=63, X
 SUBAREA=46, X
 BFRS=124, X
 USGTIER=5, X
 LTRACE=8, X
 OLT=YES, X
 DR3270=NO, X
 BRANCH=8000, X
 TRACE=(YES,100), X
 NEWNAME=EX3346Z, X
 X25.PREFIX=Z, * X
```

## Example B—IDNUM Values

```

TYPYSYS=OS, * X25 PART X
X25.SNAP=YES, * X
X25.USGTIER=5, * X
X25.MCHCNT=4, * X
X25.MWINDOW=1, * X
X25.IDNUMH=2, * X25.IDNUMH MUST MATCH WITH SWITCHX
X25.MAXPIU=64K, * X
X25.PAHINDX=2 * MAX NUMBER OF PAD PARAMETERS

*
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD, X
RIMM,NAKLIM,SESSION,SSPAUSE,XMTLMT)

*
ATTACHED HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16, X
DELAY=0.1,STATMOD=YES,TIMEOUT=30

*
PATH45 PATH DESTSA=(45),ER1=(45,2),ER2=(45,3) * CCU REMOTE
PATH16 PATH DESTSA=(16),ER0=(16,1),ER1=(16,1),ER2=(16,1)
* LUPOOL NUMBER=5
LUDRPOOL NUMTYP1=10,NUMTYP2=10

EJECT
ZDLC4 GROUP LNCTL=SDLC, X
TYPE=NCP, X
DIAL=YES, X
REPLYTO=20

ZL4 LINE ADDRESS=4, X
CLOCKNG=EXT, X
NEWSYNC=NO, X
DUPLEX=HALF, X
NRZI=NO, X
CALL=INOUT, X
RETRIES=(7,0,0)
ZP4 PU ISTATUS=INACTIVE, X
PUTYPE=2

EJECT

GRNPZ GROUP LNCTL=SDLC, X
TYPE=NCP, X
DIAL=NO, X
DISCNT=NO, X
VIRTUAL=YES, X
NPARSC=YES, X
ISTATUS=INACTIVE
LNNPZ LINE LINEFVT=NPAVFVT
PUNPZ PU PUTYPE=2,PUFVT=NPAPFVT
LUNPZ LU MAXCOLL=6,LUFVT=(NPALFVT,NPALFVT), X
VPACING=0,LOCADDR=1

EJECT
*
* PAD - PARAMETERS
*
PAD1 X25.PAD INDEX=1,PADPARM=0715080001000400
PAD2 X25.PAD INDEX=2,PADPARM=0701080001000400
*
* USER TRANSLATE

```



```

*
*
* NDF SAMPLE STATEMENTS FOR PAD USER TRANSLATE TABLES
*
TRANS1 X25.TRAN USER=1, C
 DCIN0=00010203372D2E2F1605250B0C0D0E0F, C
 DCIN1=101112133C5A322618193F271C1D1E1F, C
 DCIN2=404F7F7B5B6C507D4D5D5C4E6B604B61, C
 DCIN3=F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F, C
 DCIN4=43C1C2C3C4C5C6C7C8C9D1D2D3D4D5D6, C
 DCIN5=D7D8D9E2E3E4E5E6E7E8E9AD51BD5F6D, C
 DCIN6=79818283848586878889919293949596, C
 DCIN7=979899A2A3A4A5A6A7A8A98BBA9BA107, C
 DCIN8=80DCC083DA7C86E08FC9D08B568D8E8F, C
 DCIN9=909192FBFB95969798999A9B9C9D9E9F, C
 DCINA=A0A1A2A3A4A5A6A7A8A9AA64ACADAEAF, C
 DCINB=B0B1B2FAECB5B6B7B8B9BABBBCBDBEBC, C
 DCINC=ABCBCCEBBF8FC6C7ABC9CACBCCDCCECF, C
 DCIND=D0D1D2D3D4D5D6D7D8BBACDBDCDDDEDF, C
 DCINE=E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF, C
 DCINF=F0F1F2F3F4F5F6F7B0F9FAFBFCFDFFEF, C
 DCOT0=000102033A093A7F3A3A3A0B0C0D0E0F, C
 DCOT1=101112133A0A087F18193A3A1C1D1E1F, C
 DCOT2=3A3A3A3A3A0A171B3A3A3A3A3A050607, C
 DCOT3=3A3A163A3A3A3A043A3A3A3A14153A1A, C
 DCOT4=203A3A403A3A3A3A3A3A5B2E3C282B21, C
 DCOT5=265C3A3A3A8C3A3A3A15242A293B5E, C
 DCOT6=2D2F3A3AAB3A3A3A3A3A7C2C255F3E3F, C
 DCOT7=3A3A3A3A3A3A3A3A3A603A2385273D22, C
 DCOT8=3A6162636465666768693A7B3A3A3AC5, C
 DCOT9=3A6A6B6C6D6E6F7071723A7D3A3A3A3A, C
 DCOTA=3A7E737475767778797A3AC0DA5B3A3A, C
 DCOTB=F83A3A3A3A3A3A3A3A3A3AD9BF5D3AC4, C
 DCOTC=824142434445464748493AC1C23A3A3A, C
 DCOTD=8A4A4B4C4D4E4F505152843A813A3AFF, C
 DCOTE=873A535455565758595A3AC3B43A3A3A, C
 DCOTF=30313233343536373839B393883A3A3A C
 0 1 2 3 4 5 6 7 8 9 A B C D E F
*
*
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES, X
 INACTO=120, CLEAR SVC AFTER 120 SECONDS IF NO TRAFIC X
 RESETINO=(XX26,80XX) FOR ALL RESET WITH DIAG CODE 26
** AND ALL RESET WITH CAUSE CODE 80
**
 X25.VCCPT INDEX=1, X
 MAXPKTL=128, X
 VWINDOW=3, X
 INSLOW=(50,12)
*
 X25.VCCPT INDEX=2, X
 MAXPKTL=128, X
 VWINDOW=2, X
 INSLOW=(100,50)
*
 X25.VCCPT INDEX=3, X
 MAXPKTL=128, X
 VWINDOW=3, X

```

## Example B—IDNUM Values

```

 INSLOW=(100,50)
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2, X
 OPTFACL=420707430202
*
X25.OUFT INDEX=3, X
 OPTFACL=03690102420707430303, *REVERSE CHARGING X
 USRFILD=1234567890
*

*
 EJECT
 EJECT
**** FIRST MCH (GATE) *****

X25.MCH ADDRESS=64, X
 ANS=CONT,NCPGRP=GRZ1, X
 LCGDEF=(0,2), X
 FRMLGTH=131, X
 MWINDOW=7, X
 STATION=DTE, X
 TPTIMER=2, X
 TDTIMER=3, X
 NPRETRY=3, X
 NDRETRY=3, X
 LCN0=USED, *FOR TRANSPAC X
 XMONLNK=YES, X
 PAD=INTEG,TRAN=USER1, *8 BITS TRANSLATE TABLE 1 X
 PADINDX=1, * FOR PAD PARAMETER SELECTION X
 GATE=GENERAL, X
 LLCLIST=(LLC0,LLC4,LLC5), X
 SPAN=X2501, X
 TITIMER=1, X
 IDBLKC=064, *PCNE X
 IDBLKP=063, *PAD X
 IDBLKG=062, *GATE X
 MBITCHN=YES, X
 DBIT=YES

 X25.LCG LCGN=0
*****BNN PCNE*****
X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC0,MAXDATA=2000, X
 SPAN=X2501,VPACING=20

*** 2 SVC,S USER DEFINED IDNUM

X25.VC LCN=(1,2),VCCINDX=2,TYPE=S,CALL=INOUT,OUFINDX=2, X
 SPAN=X2501,IDNUMT=0040
*
*

*
 EJECT
**** SECOND MCH (GATE) *****
*
 1 SVC FOR SECONDARY CASUAL CONNECTION

```

## Example B—IDNUM Values

```

* 1 PVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR PRIMARY CASUAL CONNECTION

X25.MCH ADDRESS=65, X
 ANS=CONT,NCPGRP=GRZ2, X
 LCGDEF=(0,5), X
 FRMLGTH=131, X
 STATION=DTE, X
 TPTIMER=3.1, X
 TDTIMER=2, X
 NPRETRY=7, X
 NDRETRY=1, X
 LLCLIST=(LLC2,LLC3), X
 CCXDELAY=5, *INITIAL TRANSMISSION DELAY CASUAL CONNECT X
 ROLE=PRI, *LOCAL PERIPHERAL PU 2.1 ASSUMES PRIMARY ROLE X
 LSPRI=YES, *ALLOWS TRAFIC PRIORITY FOR LLC2 AND BNN LLC3X
 LCN0=USED, X
 GATE=GENERAL, *FOR A PVC X
 CUD0=(ALL), X
 INTFAC=YES, X
 MWINDOW=7, X
 TITIMER=1, X
 MBITCHN=NO

X25.LCG LCGN=0

* *** PVC 0 FOR PU2.1 ON BNN QLLC ***
* CASUAL CONNECTION (PRIMARY)
*
X25.LINE LCN=0, * X
 VCCINDX=3, * X
 LLC=LLC3, * DSTNODE=BNN DEFAULTED X
 SPAN=X2501, * X
 TYPE=P *
*
X25.PU PUTYPE=2, * X
 ADDR=01, * X
 VPACING=2, * X
 XID=YES, * LEN X
 DATMODE=HALF, * X
 PACING=1, * X
 SPAN=X2501, * X
 ISTATUS=INACTIVE, * X
 ANS=CONT *
*
SLUSEC26 X25.LU LOCADDR=0 * FOR INDEPENDANT LU
*
**** 3 SVC'S *****

X25.VC LCN=(1,3), X
 VCCINDX=3, X
 OUFINDX=1, X
 CALL=INOUT, X
 ID=06344444, * ANONYMOUS CALLER (CANNED XID) X
 TYPE=SWITCHED
*
*

```

## Example B—IDNUM Values

```

* *** PVC 3 FOR CASUAL CONNECTION (SECONDARY)
*
X25.LINE LCN=4, *
 VCCINDX=3, *
 LLC=LLC3, * QLLC
 SPAN=X2501, *
 MODE=SEC, * CCX SECONDARY DEFINITION
 TADDR=C1, * CCX SECONDARY DEFINITION
 TYPE=P *
*
X25.PU PUTYPE=2, *
 VPACING=2, *
 MODETAB=LSPRITAB, * VTAM USER DEFINED TABLE
 XID=YES, * LEN SUPPORT
 DATMODE=HALF, *
 PACING=1, *
 SPAN=X2501, *
 ISTATUS=INACTIVE, *
 ANS=CONT *
*
PLUPRI15 X25.LU LOCADDR=0, * INDEPENDANT LU
 DLOGMOD=LSPRI62B, * VTAM USER DEFINED TABLE
 ISTATUS=ACTIVE *
*

X25.VC LCN=5, *
 VCCINDX=3, *
 LLC=LLC4, *
 MAXDATA=2000, *
 VPACING=20, *
 TYPE=P *
*

*

* THIRD MCH FOR INN (REMOTE LOADING)
** RELATED WITH THE GEN NAMED NCPREM LOADED FROM DISK INTO REMOTE CCU

X25.MCH ADDRESS=66, *CONTROLLER LINE ADDR
 ANS=CONT, *
 LCGDEF=(0,1), *
 FRMLGTH=131, *
 MWINDOW=7, *LAP WINDOW
 STATION=DTE, *
 TPTIMER=3, *LAPB RECOVERY PARAMETERS
 TDTIMER=1, *
 NPRETRY=7, *
 NDRETRY=1, *
 LLCLIST=(LLC3), *
 LCN0=USED, *
 DBIT=NO, *NO DBIT SUPPORT ON THIS MCH
 ITRACE=YES, * LAP INTERNAL TRACE ACTIVE
 SPEED=9600, *REAL MCH SPEED
 TITIMER=1, *REAL T1 OF THE DCE
 SVCINN=1, *NB OF SIMULT.SWINN CONNECT.
 SDRTCNT=1, *NB OF RETRIES
 SHM=NO, *NO SHM FOR REMOTE
 SDRTIME=10 *TIMER BETWEEN RETRIES
*

```

```

X25.LCG LCGN=0
*
***** (PVC) *****
*
X25.LINE LCN=0,VCCINDX=3,LLC=LLC3,TYPE=P, X
DSTNODE=INN
*
X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE
*
***** (SVC) *****
*
X25.LINE DSTNODE=INN, *MANDATORY FOR INN X
TYPE=S
*
X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
X25.VC LCN=1,VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1
*

* FOURTH MCH FOR GATE X.21 SWITCHED ACCESS
*

X25.MCH ADDRESS=67, X
ANS=CONT, X
LCGDEF=(0,1), X
FRMLGTH=131, X
MWINDOW=5, LAP WINDOW X
STATION=DTE, X
TPTIMER=3, X
TDTIMER=2, X
NPRETRY=10, X
NDRETRY=2, X
LCN0=NOTUSED, X
GATE=GENERAL, X
SPAN=X2501, X
LLCLIST=(LLC4), X
X21SW=YES, *X21 SWITCH ACCESS TO ISDN NETWORK X
X21NTWK=1984, X
X21RTYTO=4.0, *X21 CALL OUT RETRY TIME X
X21RTYCT=5, *X21 CALL OUT RETRY COUNT X
X21INACT=40, X
SPEED=64000 SPECIFIED TO SELECT LIC4B
*
X25.LCG LCGN=0
*
** 1 SVC'S *****
*
X25.VC LCN=1, X
VCCINDX=3, X
OUFINDX=1, X
CALL=INOUT, X
IDNUMT=0080, X
TYPE=SWITCHED
*

*
*

```

## Example B—IDNUM Values

```

 X25.END
*

CZGROUP GROUP LNCTL=CA, X
 CA=TYPE6, X
 DELAY=0.0, X
 NCPCA=ACTIVE
CZLINE5 LINE ADDRESS=00
CZ5PU PU PUTYPE=5
CZLINE1 LINE ADDRESS=08
CZ1PU PU PUTYPE=5
 EJECT

 GENEND
 END

```

## NCP Input Produced by NDF (NPSI)—Example B1

```

OPTIONS USERGEN=X25NPSI,NEWDEFN=(YES,ECHO)
*
ATTACHED PCCU DUMPDS=DUMPX25,SUBAREA=16
*
BAKBUILD BUILD NPA=YES,
 VERSION=V5R3,
 LOADLIB=LNCP10,
 TYPGEN=NCP,
 MODEL=3745,
 NETID=FRIBM412,
 MAXSUBA=63,
 SUBAREA=46,
 BFRS=124,
 USGTIER=5,
 LTRACE=8,
 OLT=YES,
 DR3270=NO,
 BRANCH=8000,
 TRACE=(YES,100),
 NEWNAME=EX3346Z,
 X25.PREFIX=Z,
 TYP SYS=OS,
 X25.SNAP=YES,
 X25.USGTIER=5,
 X25.MCHCNT=4,
 X25.MWINDOW=1,
 X25.IDNUMH=2,
 X25.MAXPIU=64K,
 X25.PAHINDX=2
*
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD,RIMM,NAKLIM*,
 ,SESSION,SSPAUSE,XMTLMT)
*
ATTACHED HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16,
 DELAY=0.1,STATMOD=YES,TIMEOUT=30
*
PATH45 PATH DESTSA=(45),ER1=(45,2),ER2=(45,3) * CCU REMOTE
PATH16 PATH DESTSA=(16),ER0=(16,1),ER1=(16,1),ER2=(16,1)
*
LUPPOOL NUMBER=5
LUDRPOOL NUMTYP1=10,NUMTYP2=10

ZDLC4 GROUP LNCTL=SDLC,
 TYPE=NCP,
 DIAL=YES,
 REPLYTO=20

ZL4 LINE ADDRESS=4,
 CLOCKNG=EXT,
 NEWSYNC=NO,
 DUPLEX=HALF,
 NRZI=NO,
 CALL=INOUT,
 RETRIES=(7,0,0)
ZP4 PU ISTATUS=INACTIVE,
 PUTYPE=2

```

## Example B—IDNUM Values

```

GRNPZ GROUP LNCTL=SDLC,
 TYPE=NCP,
 DIAL=NO,
 DISCNT=NO,
 VIRTUAL=YES,
 NPARSC=YES,
 ISTATUS=INACTIVE
LNNPZ LINE LINEFVT=NPAVFVT
PUNPZ PU PUTYPE=2,PUFVT=NPAPFVT
LUNPZ LU MAXCOLL=6,LUFVT=(NPALFVT,NPALFVT),
 VPACING=0,LOCADDR=1

*
* PAD - PARAMETERS
*
PAD1 X25.PAD INDEX=1,PADPARAM=0715080001000400
PAD2 X25.PAD INDEX=2,PADPARAM=0701080001000400
*
* USER TRANSLATE
*
* NDF SAMPLE STATEMENTS FOR PAD USER TRANSLATE TABLES
*
TRANS1 X25.TRAN USER=1,
 DCIN0=00010203372D2E2F1605250B0C0D0E0F,
 DCIN1=101112133C5A322618193F271C1D1E1F,
 DCIN2=404F7F7B5B6C507D4D5D5C4E6B604B61,
 DCIN3=F0F1F2F3F4F5F6F7F8F9A5E4C7E6E6F,
 DCIN4=43C1C2C3C4C5C6C7C8C9D1D2D3D4D5D6,
 DCIN5=D7D8D9E2E3E4E5E6E7E8E9AD51BD5F6D,
 DCIN6=79818283848586878889919293949596,
 DCIN7=979899A2A3A4A5A6A7A8A98BBA9BA107,
 DCIN8=80DCC083DA7C8E08FC9D08B568D8E8F,
 DCIN9=909192FBFB95969798999A9B9C9D9E9F,
 DCINA=A0A1A2A3A4A5A6A7A8A9AA64ACADAEAF,
 DCINB=B0B1B2FAECB5B6B7B8B9BABBBCBDBEBEBC,
 DCINC=ABCBCCEBBF8FC6C7ABC9CACBCCDCECF,
 DCIND=D0D1D2D3D4D5D6D7D8BBACDBDCDDDEDF,
 DCINE=E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF,
 DCINF=F0F1F2F3F4F5F6F7B0F9FAFBFCFDFFEF,
 DCOT0=000102033A093A7F3A3A3A0B0C0D0E0F,
 DCOT1=101112133A0A087F18193A3A1C1D1E1F,
 DCOT2=3A3A3A3A3A0A171B3A3A3A3A3A050607,
 DCOT3=3A3A163A3A3A043A3A3A14153A1A,
 DCOT4=203A3A403A3A3A3A3A5B2E3C282B21,
 DCOT5=265C3A3A3A8C3A3A3A15242A293B5E,
 DCOT6=2D2F3A3AAB3A3A3A3A7C2C255F3E3F,
 DCOT7=3A3A3A3A3A3A3A3A603A2385273D22,
 DCOT8=3A6162636465666768693A7B3A3A3AC5,
 DCOT9=3A6A6B6C6D6E6F7071723A7D3A3A3A3A,
 DCOTA=3A7E737475767778797A3AC0DA5B3A3A,
 DCOTB=F83A3A3A3A3A3A3A3A3AD9BF5D3AC4,
 DCOTC=824142434445464748493AC1C23A3A3A,
 DCOTD=8A4A4B4C4D4E4F505152843A813A3AFF,
 DCOTE=873A535455565758595A3AC3B43A3A3A,
 DCOTF=30313233343536373839B393883A3A3A
*
* 0 1 2 3 4 5 6 7 8 9 A B C D E F

```



## Example B—IDNUM Values

```

*
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES,
 INACTO=120, CLEAR SVC AFTER 120 SECONDS IF NO TRAFIC
 RESETINO=(XX26,80XX), FOR ALL RESET WITH DIAG CODE 26
 DONE=YES
**
**
 AND ALL RESET WITH CAUSE CODE 80
*
X25.VCCPT INDEX=1,
 MAXPKTL=128,
 VWINDOW=3,
 INSLow=(50,12)
*
X25.VCCPT INDEX=2,
 MAXPKTL=128,
 VWINDOW=2,
 INSLow=(100,50)
*
X25.VCCPT INDEX=3,
 MAXPKTL=128,
 VWINDOW=3,
 INSLow=(100,50)
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2,
 OPTFACL=420707430202
*
X25.OUFT INDEX=3,
 OPTFACL=03690102420707430303, *REVERSE CHARGING
 USRFILD=1234567890
*

*
**** FIRST MCH ZL64(GATE) *****

X25.MCH ADDRESS=64,
 ANS=CONT,NCPGRP=GRZ1,
 LCGDEF=(0,2),
 FRMLGTH=131,
 MWINDOW=7,
 STATION=DTE, *XPAC1
 TPTIMER=2,
 TDTIMER=3,
 NPRETRY=3,
 NDRETRY=3,
 LCN0=USED, *FOR TRANSPAC
 XMONLNK=YES,
 PAD=INTEG,TRAN=USER1, *8 BITS TRANSLATE TABLE 1
 PADINDX=1, * FOR PAD PARAMETER SELECTION
 GATE=GENERAL,
 LLCLIST=(LLC0,LLC4,LLC5),
 SPAN=X2501,
 TITIMER=1,
 IDBLKC=064, *PCNE
 IDBLKP=063, *PAD
 IDBLKG=062, *GATE
 MBITCHN=YES,

```

## Example B—IDNUM Values

```

 DBIT=YES

 X25.LCG LCGN=0
*****BNN PCNE*****
 X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC0,MAXDATA=2000, *
 SPAN=X2501,VPACING=20

*** 2 SVC,S USER DEFINED IDNUM

 X25.VC LCN=(1,2),VCCINDX=2,TYPE=S,CALL=INOUT,OUFINDX=2, *
 SPAN=X2501,IDNUMT=0040
*
*

*
**** SECOND MCH (GATE) ZL65 *****
* 1 SVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR PRIMARY CASUAL CONNECTION

 X25.MCH ADDRESS=65, *
 ANS=CONT,NCPGRP=GRZ2, *
 LCGDEF=(0,5), *
 FRMLGTH=131, *
 STATION=DTE, *XPAC3 *
 TPTIMER=3.1, *
 TDTIMER=2, *
 NPRETRY=7, *
 NDRETRY=1, *
 LLCLIST=(LLC2,LLC3), *
 CCXDELAY=5, *INITIAL TRANSMISSION DELAY CASUAL CONNECT *
 ROLE=PRI, *LOCAL PERIPHERAL PU 2.1 ASSUMES PRIMARY ROLE *
 LSPRI=YES, *ALLOWS TRAFIC PRIORITY FOR LLC2 AND BNN LLC3*
 LCN0=USED, *
 GATE=GENERAL, *FOR A PVC *
 CUD0=(ALL), *
 INTFAC=YES, *
 MWINDOW=7, *
 TITIMER=1, *
 MBITCHN=NO

 X25.LCG LCGN=0

* *** PVC 0 FOR PU2.1 ON BNN QLLC ***
* CASUAL CONNECTION (PRIMARY)
*
 X25.LINE LCN=0, *
 VCCINDX=3, *
 LLC=LLC3, * DSTNODE=BNN DEFAULTED *
 SPAN=X2501, *
 TYPE=P *
*
 X25.PU PUTYPE=2, *
 ADDR=01, *
 VPACING=2, *
 XID=YES, * LEN *
 DATMODE=HALF, *

```

## Example B—IDNUM Values

```

 PACING=1, *
 SPAN=X2501, *
 ISTATUS=INACTIVE, *
 ANS=CONT *
*
SLUSEC26 X25.LU LOCADDR=0 * FOR INDEPENDANT LU
*
**** 3 SVC'S *****

 X25.VC LCN=(1,3),
 VCCINDX=3,
 OUFINDX=1,
 CALL=INOUT,
 ID=06344444, * ANONYMOUS CALLER (CANNED XID)
 TYPE=SWITCHED
*
*
* *** PVC 3 FOR CASUAL CONNECTION (SECONDARY)
*
 X25.LINE LCN=4,
 VCCINDX=3,
 LLC=LLC3, * QLLC
 SPAN=X2501,
 MODE=SEC, * CCX SECONDARY DEFINITION
 TADDR=C1, * CCX SECONDARY DEFINITION
 TYPE=P
*
 X25.PU PUTYPE=2,
 VPACING=2,
 MODETAB=LSPRITAB, * VTAM USER DEFINED TABLE
 XID=YES, * LEN SUPPORT
 DATMODE=HALF,
 PACING=1,
 SPAN=X2501,
 ISTATUS=INACTIVE,
 ANS=CONT
*
PLUPRI15 X25.LU LOCADDR=0, * INDEPENDANT LU
 DLOGMOD=LSPRI62B, * VTAM USER DEFINED TABLE
 ISTATUS=ACTIVE
*

 X25.VC LCN=5,
 VCCINDX=3,
 LLC=LLC4,
 MAXDATA=2000,
 VPACING=20,
 TYPE=P

*

* THIRD MCH FOR INN (REMOTE LOADING) ZL66
** RELATED WITH THE GEN NAMED NCPREM LOADED FROM DISK INTO REMOTE CCU

 X25.MCH ADDRESS=66, *CONTROLLER LINE ADDR
 ANS=CONT,
 LCGDEF=(0,1), *MAX LCN IS 7 ON LCGN 0
 FRMLGTH=131,

```

## Example B—IDNUM Values

```

MWINDOW=7, *LAP WINDOW *
STATION=DTE, *
TPTIMER=3, *LAPB RECOVERY PARAMETERS *
TDTIMER=1, *
NPRETRY=7, *
NDRETRY=1, *
LLCLIST=(LLC3), *
LCN0=USED, *
DBIT=NO, *NO DBIT SUPPORT ON THIS MCH *
ITRACE=YES, * LAP INTERNAL TRACE ACTIVE *
SPEED=9600, *REAL MCH SPEED *
T1TIMER=1, *REAL T1 OF THE DCE *
SVCINN=1, *NB OF SIMULT.SWINN CONNECT. *
SDRTCNT=1, *NB OF RETRIES *
SHM=NO, *NO SHM FOR REMOTE *
SDRTIME=10 *TIMER BETWEEN RETRIES *
*
X25.LCG LCGN=0
*
***** (PVC) *****
*
X25.LINE LCN=0,VCCINDX=3,LLC=LLC3,TYPE=P, *
DSTNODE=INN
*
X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE
*
***** (SVC) *****
*
X25.LINE DSTNODE=INN, *MANDATORY FOR INN *
TYPE=S
*
X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
X25.VC LCN=1,VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1
*

* FOURTH MCH FOR GATE X.21 SWITCHED ACCESS ZL67
*

X25.MCH ADDRESS=67, *
ANS=CONT, *
LCGDEF=(0,1), *
FRMLGTH=131, *
MWINDOW=5, LAP WINDOW *
STATION=DTE, *
TPTIMER=3, *
TDTIMER=2, *
NPRETRY=10, *
NDRETRY=2, *
LCN0=NOTUSED, *
GATE=GENERAL, *
SPAN=X2501, *
LLCLIST=(LLC4), *
X21SW=YES, *X21 SWITCH ACCESS TO ISDN NETWORK *
X21NTWK=1984, *
X21RTYT0=4.0, *X21 CALL OUT RETRY TIME *
X21RTYCT=5, *X21 CALL OUT RETRY COUNT *
X21INACT=40, *

```

```

 SPEED=64000 SPECIFIED TO SELECT LIC4B
*
X25.LCG LCGN=0
*
** 1 SVC'S *****
*
X25.VC LCN=1,
 VCCINDX=3,
 OUFINDX=1,
 CALL=INOUT,
 IDNUMT=0080,
 TYPE=SWITCHED
*

*
*
X25.END
* GENERATED BY X25NPSI
Z25P64A GROUP DIAL=NO,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALPBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL64000 LINE IPL=NO,
 SPAN=X2501,
 UACB=XA64000
* GENERATED BY X25NPSI
ZC64000 SERVICE
* GENERATED BY X25NPSI
ZP64000 PU ADDR=01,
 PUDR=NO,
 PUTYPE=1,
 XID=NO,
 SPAN=X2501,
 MAXDATA=2000
* GENERATED BY X25NPSI
ZU64000 LU LOCADDR=0,
 SPAN=X2501,
 VPACING=20,
 PACING=1,
 LUDR=NO
* GENERATED BY X25NPSI
Z25P65A GROUP DIAL=NO,
 LSPRI=YES,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,

```

## Example B—IDNUM Values

```
LINEADD=NONE, *
COMPTAD=NO, *
COMPOWN=YES, *
COMPSWP=NO, *
COMPACB=NO, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALNATER,,BALNATST,BALNATLS), *
TYPE=NCP, *
USERID=(5688035,BALPBDT,NORECMS), *
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65000 LINE SPAN=X2501, *
 UACB=XA65000
* GENERATED BY X25NPSI
ZC65000 SERVICE
* GENERATED BY X25NPSI
ZP65000 PU ADDR=01, *
 ANS=CONT, *
 DATMODE=HALF, *
 PACING=1, *
 ISTATUS=INACTIVE, *
 SPAN=X2501, *
 VPACING=2, *
 PUTYPE=2, *
 XID=YES, *
 PUDR=NO
* GENERATED BY X25NPSI
SLUSEC26 LU LOCADDR=0, *
 LUDR=NO
* GENERATED BY X25NPSI
Z25P65C GROUP DIAL=NO, *
 LSPRI=YES, *
 MODE=SEC, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALPBDT,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65004 LINE SPAN=X2501, *
 TADDR=C1, *
 UACB=XA65004
* GENERATED BY X25NPSI
ZC65004 SERVICE
* GENERATED BY X25NPSI
ZP65004 PU ANS=CONT, *
 DATMODE=HALF, *
 PACING=1, *
```

```

 ISTATUS=INACTIVE,
 MODETAB=LSPRITAB,
 SPAN=X2501,
 VPACING=2,
 PUTYPE=2,
 XID=YES,
 PUDR=NO
* GENERATED BY X25NPSI
PLUPRI15 LU LOCADDR=0,
 DLOGMOD=LSPRI62B,
 ISTATUS=ACTIVE,
 LUDR=NO
* GENERATED BY X25NPSI
Z25P65D GROUP DIAL=NO,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALPBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65005 LINE IPL=NO,
 UACB=XA65005
* GENERATED BY X25NPSI
ZC65005 SERVICE
* GENERATED BY X25NPSI
ZP65005 PU ADDR=01,
 PUDR=NO,
 PUTYPE=1,
 XID=NO,
 MAXDATA=2000
* GENERATED BY X25NPSI
ZU65005 LU LOCADDR=0,
 VPACING=20,
 PACING=1,
 LUDR=NO
* GENERATED BY X25NPSI
Z25P66A GROUP DIAL=NO,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,

```

## Example B—IDNUM Values

```

 USERID=(5688035,BALPBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
ZL66000 LINE UACB=XA66000
* GENERATED BY X25NPSI
ZC66000 SERVICE
* GENERATED BY X25NPSI
ZP66000 PU ANS=CONT,
 TGN=2,
 ISTATUS=INACTIVE,
 PUTYPE=4,
 PUDR=NO
* GENERATED BY X25NPSI
Z25S67A GROUP DIAL=YES,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LINEAUT=YES,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALSBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
ZL67001 LINE CALL=INOUT,
 UACB=XA67001
* GENERATED BY X25NPSI
ZP67001 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
Z25S66B GROUP DIAL=YES,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALIBDT,NORECMS),
 PUTYPE=4,
 SHM=NO,
 ACTIVTO=0.0,
 BRKCON=NONE,
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
ZL66GGH LINE CALL=INOUT,
```



```

 UACB=XA66GGH
* GENERATED BY X25NPSI
ZP66GGH PU ISTATUS=INACTIVE, *
 PUTYPE=4
* GENERATED BY X25NPSI
Z25S65B GROUP DIAL=YES, *
 LSPRI=YES, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LINEAUT=YES, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALSBTD,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
ZL65003 LINE CALL=INOUT, *
 ROLE=PRI, *
 ID=06344448, *
 UACB=XA65003
* GENERATED BY X25NPSI
ZP65003 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
ZL65002 LINE CALL=INOUT, *
 ROLE=PRI, *
 ID=06344446, *
 UACB=XA65002
* GENERATED BY X25NPSI
ZP65002 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
ZL65001 LINE CALL=INOUT, *
 ROLE=PRI, *
 ID=06344444, *
 UACB=XA65001
* GENERATED BY X25NPSI
ZP65001 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
Z25S64B GROUP DIAL=YES, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LINEAUT=YES, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *

```

## Example B—IDNUM Values

```

 USERID=(5688035,BALSBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
ZL64002 LINE CALL=INOUT,
 SPAN=X2501,
 UACB=XA64002
* GENERATED BY X25NPSI
ZP64002 PU SPAN=X2501,
 PUTYPE=(1,2)
* GENERATED BY X25NPSI
ZL64001 LINE CALL=INOUT,
 SPAN=X2501,
 UACB=XA64001
* GENERATED BY X25NPSI
ZP64001 PU SPAN=X2501,
 PUTYPE=(1,2)
* GENERATED BY X25NPSI
GRZ1 GROUP DIAL=NO,
 BERPROC=BALNUMBER,
 COMPACB=YES,
 COMPTAD=YES,
 COMPOWN=YES,
 COMPSWP=YES,
 LEVEL2=BALNAML2,
 LEVEL3=BALNAML3,
 LEVEL5=NCP,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALLAP4,,BALLAP4,BALLAP4),
 TYPE=NCP,
 USERID=(5688035,BALMBDT,NORECMS,P),
 XIO=(BALNAMXL,BALNAMXS,BALNAMXI,BALNAMXX)
* GENERATED BY X25NPSI
ZL64 LINE ADDRESS=(64,FULL),
 SPAN=X2501,
 XMONLNK=YES,
 UACB=(X25A64X,X25A64R)
* GENERATED BY X25NPSI
ZC64 SERVICE
* GENERATED BY X25NPSI
ZP64 PU ADDR=01,
 ANS=CONT,
 SPAN=X2501,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
* GENERATED BY X25NPSI
ZU64 LU LOCADDR=0,
 SPAN=X2501,
 VPACING=(2,1),
 ISTATUS=ACTIVE
* GENERATED BY X25NPSI
GRZ2 GROUP DIAL=NO,
 BERPROC=BALNUMBER,
 COMPACB=YES,
 COMPTAD=YES,
 COMPOWN=YES,
```

```

COMPSWP=YES, *
LEVEL2=BALNAML2, *
LEVEL3=BALNAML3, *
LEVEL5=NCP, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALLAP4,,BALLAP4,BALLAP4), *
TYPE=NCP, *
USERID=(5688035,BALNBDT,NORECMS,P), *
XIO=(BALNAMXL,BALNAMXS,BALNAMXI,BALNAMXK)
* GENERATED BY X25NPSI
ZL65 LINE ADDRESS=(65,FULL), *
 XMONLNK=NO, *
 UACB=(X25A65X,X25A65R)
* GENERATED BY X25NPSI
ZC65 SERVICE
* GENERATED BY X25NPSI
ZP65 PU ADDR=01, *
 ANS=CONT, *
 MAXDATA=261, *
 PUDR=NO, *
 XID=NO, *
 PUTYPE=1
* GENERATED BY X25NPSI
ZU65 LU LOCADDR=0, *
 VPACING=(2,1), *
 ISTATUS=ACTIVE
* GENERATED BY X25NPSI
ZU65L1 LU LOCADDR=1, *
 VPACING=(2,1), *
 ISTATUS=ACTIVE
* GENERATED BY X25NPSI
ZL66 LINE ADDRESS=(66,FULL), *
 XMONLNK=NO, *
 UACB=(X25A66X,X25A66R)
* GENERATED BY X25NPSI
ZC66 SERVICE
* GENERATED BY X25NPSI
ZP66 PU ADDR=01, *
 ANS=CONT, *
 MAXDATA=261, *
 PUDR=NO, *
 XID=NO, *
 PUTYPE=1
* GENERATED BY X25NPSI
ZU66 LU LOCADDR=0, *
 VPACING=(2,1), *
 ISTATUS=INACTIVE
* GENERATED BY X25NPSI
ZL67 LINE ADDRESS=(67,FULL), *
 SPAN=X2501, *
 SPEED=55855, *
 XMONLNK=NO, *
 UACB=(X25A67X,X25A67R)
* GENERATED BY X25NPSI
ZC67 SERVICE
* GENERATED BY X25NPSI
ZP67 PU ADDR=01, *

```

## Example B—IDNUM Values

```

 ANS=CONT,
 SPAN=X2501,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
* GENERATED BY X25NPSI
ZU67 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
*

CZGROUP GROUP LNCTL=CA,
 CA=TYPE6,
 DELAY=0.0,
 NCPCA=ACTIVE
CZLINE5 LINE ADDRESS=00
CZ5PU PU PUTYPE=5
CZLINE1 LINE ADDRESS=08
CZ1PU PU PUTYPE=5

 GENEND INIT=BALINIMD,
 TMRTICK=BALTICK,
 UGLOBAL=BALNMGOP

```

### VTAM Switched Major Node—Example B1

```

/** THIS JCL UPDATES THE SYS1.VTAMLST (MVS6)
/** GET THE FILE TO BE UPDATED UNDER THE ./ ADD CHANGE ITS NAME.
//UPD EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DISP=SHR,DSN=SYS1.VTAMLST,VOL=SER=MVSVM3,UNIT=3350
//SYSIN DD DATA
// ADD NAME=EX3346MN MEMBER NAME

* SWITCHED MAJOR NODE TO BE USED IN CORRELATION WITH EX3346Z GEN

**
EX3346MN VBUILD MAXGRP=20, *
 MAXNO=20, *
 TYPE=SWNET
**

*
* ENTRIES FOR GATE *
* FOR GATE, CALLIN IS ALWAYS SIMULATED TO VTAM. PATH NOT NEEDED *
*
* TO USE WITH GEN: EX3346Z, MCH:ZL64-ZP64-ZU64 *
* SVC:ZL64001 USER DEFINED IDNUM:06220042 (CALLIN GATEN <<<<<I *
* SVC:ZL64002 USER DEFINED IDNUM:06220044 (CALLOUT GATEN-----I *
*

*
GATEPI PU ADDR=01, GATE INCALL *
 IDBLK=062, 1ST GATE VC (INCOMING CALL) *
 IDNUM=20042, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
GATELI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*
**
GATEPO PU ADDR=01, 2ND GATE VC USED(CALL OUT WITHOUT PATH) *
 IDBLK=062, *
 IDNUM=20044, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
GATELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES

*
* ENTRIES FOR X21 GATE *
* FOR GATE, CALLIN IS ALWAYS SIMULATED TO VTAM. PATH NOT NEEDED *
* SVC:ZL67001 USER DEFINED IDNUM:00320082 *

```

## Example B—IDNUM Values

```

*

*
BATEPO PU ADDR=01, 2ND GATE VC USED(IN OR OUT CALL) *
 IDBLK=003, DEFAULT VALUE (IDBLKG NOT CODED IN GEN) *
 IDNUM=20082,
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2,
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1,
 SSCPFM=USSSCS
BATELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES *
*

*
* ENTRIES FOR PAD
*
* SVC:ZL64001 USER DEFINED IDNUM:06320042 (CALLIN PAD) *
* SVC:ZL64002 USER DEFINED IDNUM:06320044 (CALLIN CALLOUT PAD) *
*

*
PADIPI PU ADDR=01, PADI INCALL OR 1ST GATE VC USED *
 IDBLK=063, (IN OR OUT CALL) *
 IDNUM=20044,
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2,
 MAXPATH=1,
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1,
 SSCPFM=USSSCS
PADILI LU LOCADDR=0, 0 IS REQUIRED FOR NON SNA X25 DEVICES *
 SSCPFM=USSNTO, *LOGON ENDS WITH CRLF *
 MODETAB=MODRAL, *MODETAB FOR LU1 3767/TWX *
 DLOGMOD=SSCICSF, *FLIPFLOP OR CONTENTION PROT *
 TERM=TWX, *MUST BE DEFINED FOR TWX/3101 *
 USSTAB=USSTWX *SPECIAL USSTAB FOR TWX/3101 *
*
*
PADIPO PU ADDR=01, PADI INCALL OR 1ST GATE VC USED *
 IDBLK=063, (IN OR OUT CALL) *
 IDNUM=20042,
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2,
 MAXPATH=1,
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1,
 SSCPFM=USSSCS
*
PADPATH PATH DIALNO=10605023650201*20042, L=5 PAD (MODEM PAD) *

```

## Example B—IDNUM Values

```

GRPNM=Z25S64B,GID=2,PID=31 Z25S64B=GROUP OF SWITCHED
*
PADILO LU LOCADDR=0, *FOR NON SNA X25 DEVICES *
 SSCPFM=USSNTO, *LOGON ENDS WITH CRLF *
 MODETAB=MODRAL, *MODETAB FOR LU1 3767/TWX *
 DLOGMOD=SSCICSF, *FLIPFLOP OR CONTENTION PROT *
 TERM=TWX, *MUST BE DEFINED FOR TWX/3101 *
 USSTAB=USSTWX *SPECIAL USSTAB FOR TWX/3101 *

* ENTRIES FOR PCNE
* SVC:ZL64001 IDNUM=ZZZZ OF CUD FIELD *
* SVC:ZL64002 *

*
PCNEPI PU ADDR=01, PCNE TO PCNE INBOUND *
 IDBLK=064, *
 IDNUM=22225, IDNUM=ZZZZ OF INCOM CALL CUD FIELD *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXOUT=1, TO LET LOGON MSG IN AFTER ACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNELI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
**
PCNEPO PU ADDR=01, PCNE TO PCNE OUTBOUND *
 IDBLK=064, *
 IDNUM=22224, EQUAL ZZZZ *
 MAXPATH=6, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
**
PCNEP1 PATH DIALNO=10604016500201*22224, ZZZZ+1=CUD FIELD OF CALLRQ*
 GRPNM=Z25S64B,GID=2,PID=20 Z25S64B=GROUP OF SWITCHED
*
 VC'S ON INVOLVED MCH IN GENERATION

*
* ENTRY FOR INCOMING AND OUTGOING CALLS FOR BNN PSH AND QLLC*
* WITH CANNED XID. GEN=EX3346Z MCH=65 (ZL65 ZP65) *
* ID=0634444 *
* 6 *
* 8 *

PSHPO PU ADDR=01, *
 IDBLK=063, *
 IDNUM=41264, IDNUM OF REMOTE PU *
 MAXPATH=3, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=265, SEGMENTING SUPPORTED *
 PACING=3, *

```

## Example B—IDNUM Values

```

ANS=CONT, FOR SW SESSION CONTINUATION *
VPACING=3, *
PUTYPE=2, *
SSCPFM=USSSCS OUTGOING CALL FOR PSH
PSHP PATH DIALNO=10604016420201, L=2 FOR PSH MODEM PTT1*
GRPNM=Z25S65B,GID=2,PID=31 Z25S65B=GROUP OF SWITCHED
*
PSHLO LU LOCADDR=2,MODETAB=MODTD3
*
PSHPI PU ADDR=01, *
IDBLK=063, *
IDNUM=44444, IDNUM FOR ANONYMOUS CALLING PU *
MAXPATH=3, *
DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
MAXDATA=265, SEGMENTING SUPPORTED *
PACING=3, *
ANS=CONT, FOR SW SESSION CONTINUATION *
VPACING=3, *
PUTYPE=2, *
SSCPFM=USSSCS
PSHLI LU LOCADDR=2,MODETAB=MODTD3
*
QBNNPO PU ADDR=01, *
IDBLK=063, *
IDNUM=44446, IDNUM FOR ANONYMOUS CALLING PU *
MAXPATH=3, *
DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
MAXDATA=265, SEGMENTING SUPPORTED *
PACING=3, *
ANS=CONT, FOR SW SESSION CONTINUATION *
VPACING=3, *
PUTYPE=2, *
SSCPFM=USSSCS
QBNNP PATH DIALNO=10604016530101, L=3 FOR BNN QLLC MODEM PTT2*
GRPNM=Z25S65B,GID=2,PID=32 Z25S65B=GROUP OF SWITCHED
*
QBNNLO LU LOCADDR=2,MODETAB=MODTD3
PSHP12 PU ADDR=01, *
IDBLK=063, *
IDNUM=44448, IDNUM FOR ANONYMOUS CALLING PU *
MAXPATH=3, *
DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
MAXDATA=265, SEGMENTING SUPPORTED *
PACING=3, *
ANS=CONT, FOR SW SESSION CONTINUATION *
VPACING=3, *
PUTYPE=2, *
SSCPFM=USSSCS
PSHLI2 LU LOCADDR=2,MODETAB=MODTD3

*
* ENTRY FOR SWITCHED CASUAL CONNECTION *
* - IFNYL7P IS A CEN(VTAM/NCP) WORKING AS A NODE T2.1*

*
IFNYL7P PU ADDR=01, *
MAXPATH=1, *
DISCNT=YES, *

```



## Example B—IDNUM Values

```

 PACING=1, *
 VPACING=2, *
 PUTYPE=2, *
 ANS=CONT, *
 MODETAB=LSPRITAB, VTAM USER DEFINED TABLE *
 SSCPFM=USSSCS, *
 CPNAME=IFNYL7 ADJACENT LINK STATION SSCP NAME
 (REPLACES IDNUM/IDBLK FOR PUT2.1)
*
**
IFNYL7P1 PATH DIALNO=10604016930201, (ALS MODEM) *
 GRPNM=Z25S65B, *
 GID=2, *
 PID=33
*
SLUSEC26 LU LOCADDR=0, IDEPENDANT LU DEFINITION (NEEDED FOR CCX) *
 DLOGMOD=LSPRI62B VTAM USER DEFINED TABLE

*
* ENTRY FOR REMOTE LOAD

*
*
* VTAM....NCP SA46-----| X |-----NCP SA45
* | 25 |
*

*
* SWITCHED MAJOR NODE FOR X25 FOR BOTH NCP'S WHICH ARE: *
* - SA46 LOCAL *
* - SA45 REMOTE (RELATED REMOTE GEN IS REMNCP) *
*

**
** FOR OUTGOING CALL ISSUED FROM LOCAL TO REMOTE
*
SADREM PU PUTYPE=4, * LOADABLE REMOTE PU *
 ADDR=01, * VALUE ANY (OPERAND IGNORED FOR SAD CONN.) *
 TGN=3, *
 SUBAREA=45, *
 IDNUM=AAAAA, *
 IDBLK=003, *
 NETID=FRIBM412, *
 ANS=CONT, *
 MAXPATH=1, *
 MAXDATA=1024
*
LOCPH PATH DIALNO=10604016410201, L=1 FOR INN SWITCHED MODEM PTT1*
 GRPNM=Z25S66B,PID=34
*
SADLOC PU PUTYPE=4, * PU LOADER *
 ADDR=01, * VALUE ANY (OPERAND IGNORED FOR SAD CONN.) *
 TGN=3, *
 SUBAREA=46, *
 IDNUM=AAAAA, *
 IDBLK=003, *
 NETID=FRIBM412, *
 ANS=CONT, *

```

## Example B—IDNUM Values

```
MAXPATH=1, *
MAXDATA=1024

```

## NCP and NPSI Generation Input—Example B2

Example B2 shows a configuration similar to example B1. However, this example illustrates the use of user-defined IDNUM values.

```

*
*
* - MCH1 (64) PVC0 FOR PCNE *
* 2SVC FOR PCNE PAD GATE *
* - MCH2 (65) PVC0 FOR CASUAL CONNECT *
* 3SVC FOR PSH,BNNQLLC AND CASUAL CONNECT *
* PVC4 FOR CASUAL CONNECT *
* PVC5 FOR GATE *
* - MCH3 (66) PVC0 FOR INN QLLC REMOTE LOADING *
* SVC1 FOR INN QLLC REMOTE LOADING *
* - MCH4 (67) SVC1 FOR X21 ACCESS (GATE) *

*IDNUM ARE DEFAULT VALUES (NO IDNUMT SPECIFIED IN X25.VC/X25.LINE) *

*
**
**
*
OPTIONS USERGEN=X25NPSI,NEWDEFN=(YES,ECHO)
*
ATTACHED PCCU DUMPDS=DUMPX25,SUBAREA=16
*
BAKBUILD BUILD NPA=YES, X
VERSION=V5R3, X
LOADLIB=LNCP10, X
TYPGEN=NCP, X
MODEL=3745, X
NETID=FRIBM412, X
MAXSUBA=63, X
SUBAREA=46, X
BFRS=124, X
USGTIER=5, X
LTRACE=8, X
OLT=YES, X
DR3270=NO, X
BRANCH=8000, X
TRACE=(YES,100), X
NEWNAME=PT3346Z, X
X25.PREFIX=Z, * X
TYP SYS=OS, * X25 PART X
X25.SNAP=YES, * X
X25.USGTIER=5, * X
X25.MCHCNT=4, * X
X25.MWINDOW=1, * X
X25.IDNUMH=2, * X25.IDNUMH MUST MATCH WITH SWITCHX
X25.MAXPIU=64K, * X
```

## Example B—IDNUM Values

```

X25.PAHINDX=2 * MAX NUMBER OF PAD PARAMETERS
*
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD, X
 RIMM,NAKLIM,SESSION,SSPAUSE,XMTLMT)
*
ATTACHED HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16, X
 DELAY=0.1,STATMOD=YES,TIMEOUT=30
*
PATH45 PATH DESTSA=(45),ER1=(45,2),ER2=(45,3)
PATH16 PATH DESTSA=(16),ER0=(16,1),ER1=(16,1),ER2=(16,1)
*
LUPPOOL NUMBER=5
LUDRPOOL NUMTYP1=10,NUMTYP2=10

EJECT
ZDLC4 GROUP LNCTL=SDLC, X
 TYPE=NCP, X
 DIAL=YES, X
 REPLYTO=20

ZL4 LINE ADDRESS=4, X
 CLOCKNG=EXT, X
 NEWSYNC=NO, X
 DUPLEX=HALF, X
 NRZI=NO, X
 CALL=INOUT, X
 RETRIES=(7,0,0)
ZP4 PU ISTATUS=INACTIVE, X
 PUTYPE=2
*
EJECT

GRNPZ GROUP LNCTL=SDLC, X
 TYPE=NCP, X
 DIAL=NO, X
 DISCNT=NO, X
 VIRTUAL=YES, X
 NPARSC=YES, X
 ISTATUS=INACTIVE
LNNPZ LINE LINEFVT=NPAVFVT
PUNPZ PU PUTYPE=2,PUFVT=NPAPFVT
LUNPZ LU MAXCOLL=7,LUFVT=(NPALFVT,NPALFVT), X
 VPACING=0,LOCADDR=1

EJECT
*
* PAD - PARAMETERS
*
PAD1 X25.PAD INDEX=1,PADPARM=0715080001000400
PAD2 X25.PAD INDEX=2,PADPARM=0701080001000400
*
* USER TRANSLATE
*
*
* NDF SAMPLE STATEMENTS FOR PAD USER TRANSLATE TABLES
*
TRANS1 X25.TRAN USER=1, C
 DCIN0=00010203372D2E2F1605250B0C0D0E0F, C
 DCIN1=101112133C5A322618193F271C1D1E1F, C

```

## Example B—IDNUM Values

```

DCIN2=404F7F7B5B6C507D4D5D5C4E6B604B61, C
DCIN3=F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F, C
DCIN4=43C1C2C3C4C5C6C7C8C9D1D2D3D4D5D6, C
DCIN5=D7D8D9E2E3E4E5E6E7E8E9AD51BD5F6D, C
DCIN6=79818283848586878889919293949596, C
DCIN7=979899A2A3A4A5A6A7A8A98BBA9BA107, C
DCIN8=80DCC083DA7C86E08FC9D08B568D8E8F, C
DCIN9=909192FBFB95969798999A9B9C9D9E9F, C
DCINA=A0A1A2A3A4A5A6A7A8A9AA64ACADAEAF, C
DCINB=B0B1B2FAECB5B6B7B8B9BABBBCBDBEBC, C
DCINC=ABCBCCEBBF8FC6C7ABC9CACBCCDCECF, C
DCIND=D0D1D2D3D4D5D6D7D8BBACBDCDDDEDF, C
DCINE=E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF, C
DCINF=F0F1F2F3F4F5F6F7B0F9FAFBFCFDFFEF, C
DCOT0=000102033A093A7F3A3A3A0B0C0D0E0F, C
DCOT1=101112133A0A087F18193A3A1C1D1E1F, C
DCOT2=3A3A3A3A3A0A171B3A3A3A3A050607, C
DCOT3=3A3A163A3A3A043A3A3A14153A1A, C
DCOT4=203A3A403A3A3A3A3A5B2E3C282B21, C
DCOT5=265C3A3A3A8C3A3A15242A293B5E, C
DCOT6=2D2F3A3AAB3A3A3A3A7C2C255F3E3F, C
DCOT7=3A3A3A3A3A3A3A3A603A2385273D22, C
DCOT8=3A6162636465666768693A7B3A3A3AC5, C
DCOT9=3A6A6B6C6D6E6F7071723A7D3A3A3A, C
DCOTA=3A7E737475767778797A3AC0DA5B3A3A, C
DCOTB=F83A3A3A3A3A3A3A3A3AD9BF5D3AC4, C
DCOTC=824142434445464748493AC1C23A3A3A, C
DCOTD=8A4A4B4C4D4E4F505152843A813A3AFF, C
DCOTE=873A535455565758595A3AC3B43A3A3A, C
DCOTF=30313233343536373839B393883A3A3A
* 0 1 2 3 4 5 6 7 8 9 A B C D E F
*
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES, X
 INACTO=120, CLEAR SVC AFTER 120 SECONDS IF NO TRAFIC X
 RESETINO=(XX26,80XX) FOR ALL RESET WITH DIAG CODE 26 X
** AND ALL RESET WITH CAUSE CODE 80
**
X25.VCCPT INDEX=1, X
 MAXPKTL=128, X
 VWINDOW=3, X
 INSLOW=(50,12)
*
X25.VCCPT INDEX=2, X
 MAXPKTL=128, X
 VWINDOW=2, X
 INSLOW=(100,50)
*
X25.VCCPT INDEX=3, X
 MAXPKTL=128, X
 VWINDOW=3, X
 INSLOW=(100,50)
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2, X
 OPTFACL=420707430202
*
X25.OUFT INDEX=3, X

```

## Example B—IDNUM Values

```

OPTFACL=03690102420707430303, *REVERSE CHARGING X
USRFILD=1234567890

*

*
 EJECT
 EJECT
**** FIRST MCH ZL64(GATE) *****

 X25.MCH ADDRESS=64, X
 ANS=CONT,NCPGRP=GRZ1, X
 LCGDEF=(0,2), X
 FRMLGTH=131, X
 MWINDOW=7, X
 STATION=DTE, X
 TPTIMER=2, X
 TDTIMER=3, X
 NPRETRY=3, X
 NDRETRY=3, X
 LCN0=USED, *FOR TRANSPAC X
 XMONLNK=YES, X
 PAD=INTEG,TRAN=USER1 *8 BITS TRANSLATE TABLE 1 X
 PADINDX=1, * FOR PAD PARAMETER SELECTION X
 GATE=GENERAL, X
 LLCLIST=(LLC0,LLC4,LLC5) X
 SPAN=X2501, X
 T1TIMER=1, X
 IDBLKC=064, *PCNE X
 IDBLKP=063, *PAD X
 IDBLKG=062, *GATE X
 MBITCHN=YES, X
 DBIT=YES

 X25.LCG LCGN=0
*****BNN PCNE*****
 X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC0,MAXDATA=2000, X
 SPAN=X2501,VPACING=20

*** 2 SVC,S DEFAULT IDNUM

 X25.VC LCN=(1,2),VCCINDX=2,TYPE=S,CALL=INOUT,OUFINDX=2, X
 SPAN=X2501

*
*

*
 EJECT
**** SECOND MCH (GATE) ZL65 *****
*
* THIS MCH INCLUDES DEFINITIONS FOR:
* 1 SVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR PRIMARY CASUAL CONNECTION

 X25.MCH ADDRESS=65, X
 ANS=CONT,NCPGRP=GRZ2, X
 LCGDEF=(0,5), X
 FRMLGTH=131, X

```

## Example B—IDNUM Values

```

STATION=DTE, X
TPTIMER=3.1, X
TDTIMER=2, X
NPRETRY=7, X
NDRETRY=1, X
LLCLIST=(LLC2,LLC3), X
CCXDELAY=5, *INITIAL TRANSMISSION DELAY CASUAL CONNECT X
ROLE=PRI, *LOCAL PERIPHERAL PU 2.1 ASSUMES PRIMARY ROLE X
LSPRI=YES, *ALLOWS TRAFIC PRIORITY FOR LLC2 AND BNN LLC3X
LCN0=USED, X
GATE=GENERAL, *FOR A PVC X
CUD0=(ALL), X
INTFAC=YES, X
MWINDOW=7, X
TITIMER=1, X
MBITCHN=NO

X25.LCG LCGN=0

* *** PVC 0 FOR PU2.1 ON BNN QLLC ***
* CASUAL CONNECTION (PRIMARY)
*
X25.LINE LCN=0, * X
VCCINDX=3, * X
LLC=LLC3, * DSTNODE=BNN DEFAULTED X
SPAN=X2501, * X
TYPE=P *
*
X25.PU PUTYPE=2, * X
ADDR=01, * X
VPACING=2, * X
XID=YES, * LEN X
DATMODE=HALF, * X
PACING=1, * X
SPAN=X2501, * X
ISTATUS=INACTIVE, * X
ANS=CONT *
*
SLUSEC26 X25.LU LOCADDR=0 * FOR INDEPENDANT LU
*
**** 3 SVC'S *****

X25.VC LCN=(1,3), X
VCCINDX=3, X
OUFINDX=1, X
CALL=INOUT, X
ID=06344444, * ANONYMOUS CALLER (CANNED XID) X
TYPE=SWITCHED
*
*
* *** PVC 3 FOR CASUAL CONNECTION (SECONDARY)
*
X25.LINE LCN=4, * X
VCCINDX=3, * X
LLC=LLC3, * QLLC X
SPAN=X2501, * X
MODE=SEC, * CCX SECONDARY DEFINITION X
TADDR=C1, * CCX SECONDARY DEFINITION X

```

## Example B—IDNUM Values

```

 TYPE=P *
*
X25.PU PUTYPE=2, * X
 VPACING=2, * X
 MODETAB=LSPRITAB, * VTAM USER DEFINED TABLE X
 XID=YES, * LEN SUPPORT X
 DATMODE=HALF, * X
 PACING=1, * X
 SPAN=X2501, * X
 ISTATUS=INACTIVE, * X
 ANS=CONT *
*
PLUPRI15 X25.LU LOCADDR=0, * INDEPENDANT LU X
 DLOGMOD=LSPRI62B, * VTAM USER DEFINED TABLE X
 ISTATUS=ACTIVE *
*

X25.VC LCN=5, X
 VCCINDX=3, X
 LLC=LLC4, X
 MAXDATA=2000, X
 VPACING=20, X
 TYPE=P

*

* THIRD MCH FOR INN (REMOTE LOADING) ZL66
** RELATED WITH THE GEN NAMED NCPREM LOADED FROM DISK INTO REMOTE CCU

X25.MCH ADDRESS=66, *CONTROLLER LINE ADDR X
 ANS=CONT, X
 LCGDEF=(0,1), X
 FRMLGTH=131, X
 MWINDOW=7, *LAP WINDOW X
 STATION=DTE, X
 TPTIMER=3, *LAPB RECOVERY PARAMETERS X
 TDTIMER=1, * X
 NPRETRY=7, * X
 NDRETRY=1, * X
 LLCLIST=(LLC3), X
 LCN0=USED, X
 DBIT=NO, *NO DBIT SUPPORT ON THIS MCH X
 ITRACE=YES, * LAP INTERNAL TRACE ACTIVE X
 SPEED=9600, *REAL MCH SPEED X
 T1TIMER=1, *REAL T1 OF THE DCE X
 SVCINN=1, *NB OF SIMULT.SWINN CONNECT. X
 SDRTCNT=1, *NB OF RETRIES X
 SHM=NO, *NO SHM FOR REMOTE X
 SDRTIME=10 *TIMER BETWEEN RETRIES
*
X25.LCG LCGN=0
*
***** (PVC) *****
*
X25.LINE LCN=0,VCCINDX=3,LLC=LLC3,TYPE=P, X
 DSTNODE=INN
*

```

## Example B—IDNUM Values

```

X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE
*
***** (SVC) *****
*
X25.LINE DSTNODE=INN, *MANDATORY FOR INN X
 TYPE=S
*
X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
X25.VC LCN=1,VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1
*

* FOURTH MCH FOR GATE X.21 SWITCHED ACCESS ZL67
*

X25.MCH ADDRESS=67, X
 ANS=CONT, X
 LCGDEF=(0,1), X
 FRMLGTH=131, X
 MWINDOW=5, LAP WINDOW X
 STATION=DTE, X
 TPTIMER=3, X
 TDTIMER=2, X
 NPRETRY=10, X
 NDRETRY=2, X
 LCN0=NOTUSED, X
 GATE=GENERAL, X
 SPAN=X2501, X
 LLCLIST=(LLC4), X
 X21SW=YES, *X21 SWITCH ACCESS TO ISDN NETWORK X
 X21NTWK=1984, X
 X21RTYT0=4.0, *X21 CALL OUT RETRY TIME X
 X21RTYCT=5, *X21 CALL OUT RETRY COUNT X
 X21INACT=40, X
 SPEED=64000 SPECIFIED TO SELECT LIC4B
*
X25.LCG LCGN=0
*
** 1 SVC'S *****
*
X25.VC LCN=1, X
 VCCINDX=3, X
 OUFINDX=1, X
 CALL=INOUT, X
 TYPE=SWITCHED
*

*
*
X25.END
*

CZGROUP GROUP LNCTL=CA, X
 CA=TYPE6, X
 DELAY=0.0, X
 NCPCA=ACTIVE
CZLINE5 LINE ADDRESS=00
CZ5PU PU PUTYPE=5

```



```
CZLINE1 LINE ADDRESS=08
CZ1PU PU PUTYPE=5
 EJECT

 GENEND
 END

```



## Example B—IDNUM Values

```

DCINF=F0F1F2F3F4F5F6F7B0F9FAFBFCFDFEFF, *
DCOT0=000102033A093A7F3A3A3A0B0C0D0E0F, *
DCOT1=101112133A0A087F18193A3A1C1D1E1F, *
DCOT2=3A3A3A3A3A0A171B3A3A3A3A050607, *
DCOT3=3A3A163A3A3A043A3A3A14153A1A, *
DCOT4=203A3A403A3A3A3A3A5B2E3C282B21, *
DCOT5=265C3A3A3A8C3A3A15242A293B5E, *
DCOT6=2D2F3A3AAB3A3A3A3A7C2C255F3E3F, *
DCOT7=3A3A3A3A3A3A3A3A603A2385273D22, *
DCOT8=3A6162636465666768693A7B3A3AC5, *
DCOT9=3A6A6B6C6D6E6F7071723A7D3A3A3A, *
DCOTA=3A7E737475767778797A3AC0DA5B3A3A, *
DCOTB=F83A3A3A3A3A3A3A3AD9BF5D3AC4, *
DCOTC=824142434445464748493AC1C23A3A3A, *
DCOTD=8A4A4B4C4D4E4F505152843A813A3AFF, *
DCOTE=873A535455565758595A3AC3B43A3A3A, *
DCOTF=30313233343536373839B393883A3A3A
*
* 0 1 2 3 4 5 6 7 8 9 A B C D E F
*
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD,RIMM,NAKLIM*,
*,SESSION,SSPAUSE,XMTLMT)
*
ATTACHED HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16, *
*,DELAY=0.1,STATMOD=YES,TIMEOUT=30
*
PATH45 PATH DESTSA=(45),ER1=(45,2),ER2=(45,3)
PATH16 PATH DESTSA=(16),ER0=(16,1),ER1=(16,1),ER2=(16,1)
*
LUPPOOL NUMBER=5
LUDRPOOL NUMTYP1=10,NUMTYP2=10

ZDLC4 GROUP LNCTL=SDLC, *
*,TYPE=NCP, *
*,DIAL=YES, *
*,REPLYTO=20

ZL4 LINE ADDRESS=4, SDLC LINE FOR NIA FRONTAL *
*,CLOCKNG=EXT, *
*,NEWSYNC=NO, *
*,DUPLEX=HALF, *
*,NRZI=NO, *
*,CALL=INOUT, *
*,RETRIES=(7,0,0)
ZP4 PU ISTATUS=INACTIVE, *
*,PUTYPE=2

ZDLC5 GROUP LNCTL=SDLC, *
*,TYPE=NCP, *
*,DIAL=NO, *
*,REPLYTO=3, *
*,DISCNT=NO, *
*,ISTATUS=INACTIVE

ZL5 LINE ADDRESS=5, LEASED SDLC MULTIPOINT LINE *
*,CLOCKNG=EXT, *
*,NEWSYNC=NO, *
*,DUPLEX=HALF, *
*,NRZI=NO,NPACOLL=YES, *
*,RETRIES=(3)

```

## Example B—IDNUM Values

```

SERVICE ORDER=(ZP5)
ZP5 PU ADDR=C2, ADDRESS 3274 *
 MAXDATA=261, *
 MAXOUT=3, *
 PUTYPE=2,NPACOLL=YES, *
 VPACING=2, *
 PACING=1 *
ZU5 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,NPACOLL=YES

GRNPZ GROUP LNCTL=SDLC, *
 TYPE=NCP, *
 DIAL=NO, *
 DISCNT=NO, *
 VIRTUAL=YES, *
 NPARSC=YES, *
 ISTATUS=INACTIVE *
LNNPZ LINE LINEFVT=NPAVFVT
PUNPZ PU PUTYPE=2,PUFVT=NPAPFVT
LUNPZ LU MAXCOLL=7,LUFVT=(NPALFVT,NPALFVT), *
 VPACING=0,LOCADDR=1

*
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES, *
 INACTO=120, CLEAR SVC AFTER 120 SECONDS IF NO TRAFIC *
 RESETINO=(XX26,80XX), FOR ALL RESET WITH DIAG CODE 26 *
 DONE=YES
**
**
 AND ALL RESET WITH CAUSE CODE 80
*
X25.VCCPT INDEX=1, *
 MAXPKTL=128, *
 VWINDOW=3, *
 INSLow=(50,12)
*
X25.VCCPT INDEX=2, *
 MAXPKTL=128, *
 VWINDOW=2, *
 INSLow=(100,50)
*
X25.VCCPT INDEX=3, *
 MAXPKTL=128, *
 VWINDOW=3, *
 INSLow=(100,50)
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2, *
 OPTFACL=420707430202
*
X25.OUFT INDEX=3, *
 OPTFACL=03690102420707430303, *REVERSE CHARGING *
 USRFILD=1234567890
*

*
**** FIRST MCH ZL64(GATE) *****

X25.MCH ADDRESS=64, *
 ANS=CONT,NCPGRP=GRZ1, *

```

## Example B—IDNUM Values

```

LCGDEF=(0,2), *
FRMLGTH=131, *
MWINDOW=7, *
STATION=DTE, *XPAC1 *
TPTIMER=2, *
TDTIMER=3, *
NPRETRY=3, *
NDRETRY=3, *
LCN0=USED, *FOR TRANSPAC *
XMONLNK=YES, *
PAD=INTEG,TRAN=USER1, *8 BITS TRANSLATE TABLE 1 *
PADINDX=1, * FOR PAD PARAMETER SELECTION *
GATE=GENERAL, *
LLCLIST=(LLC0,LLC4,LLC5), *
SPAN=X2501, *
TITIMER=1, *
IDBLKC=064, *PCNE *
IDBLKP=063, *PAD *
IDBLKG=062, *GATE *
MBITCHN=NO, *
DBIT=NO *

X25.LCG LCGN=0
*****BNN PCNE*****
X25.VC LCN=0,VCCINDX=3,TYPE=P,LLC=LLC0,MAXDATA=2000, *
SPAN=X2501,VPACING=20

*** 2 SVC,S DEFAULT IDNUM

X25.VC LCN=(1,2),VCCINDX=2,TYPE=S,CALL=INOUT,OUFINDX=2, *
SPAN=X2501

*
*

*
**** SECOND MCH (GATE) ZL65 *****
* 1 SVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR SECONDARY CASUAL CONNECTION
* 1 PVC FOR PRIMARY CASUAL CONNECTION

X25.MCH ADDRESS=65, *
ANS=CONT,NCPGRP=GRZ2, *
LCGDEF=(0,5), *
FRMLGTH=131, *
STATION=DTE, *XPAC3 *
TPTIMER=3.1, *
TDTIMER=2, *
NPRETRY=7, *
NDRETRY=1, *
LLCLIST=(LLC2,LLC3), *
CCXDELAY=5, * INITIAL TRANSMISSION DELAY CASUAL CONNECT *
ROLE=PRI, *LOCAL PERIPHERAL PU 2.1 ASSUMES PRIMARY ROLE *
LSPRI=YES, *ALLOWS TRAFIC PRIORITY FOR LLC2 AND BNN LLC3*
LCN0=USED, *
GATE=GENERAL, *FOR A PVC *
CUD0=(ALL), *
INTFAC=YES, *

```

## Example B—IDNUM Values

```

 MWINDOW=7, *
 TITIMER=1, *
 MBITCHN=NO

 X25.LCG LCGN=0

* *** PVC 0 FOR PU2.1 ON BNN QLLC ***
* CASUAL CONNECTION (PRIMARY)
*
 X25.LINE LCN=0, *
 VCCINDX=3, *
 LLC=LLC3, * DSTNODE=BNN DEFAULTED
 SPAN=X2501, *
 TYPE=P *
*
 X25.PU PUTYPE=2, *
 ADDR=01, *
 VPACING=2, *
 XID=YES, * LEN
 DATMODE=HALF, *
 PACING=1, *
 SPAN=X2501, *
 ISTATUS=INACTIVE, *
 ANS=CONT *
*
 SLUSEC26 X25.LU LOCADDR=0 * FOR INDEPENDANT LU
*
**** 3 SVC'S *****

 X25.VC LCN=(1,3), *
 VCCINDX=3, *
 OUFINDX=1, *
 CALL=INOUT, *
 ID=06344444, * ANONYMOUS CALLER (CANNED XID)
 TYPE=SWITCHED
*
*
* *** PVC 3 FOR CASUAL CONNECTION (SECONDARY)
*
 X25.LINE LCN=4, *
 VCCINDX=3, *
 LLC=LLC3, * QLLC
 SPAN=X2501, *
 MODE=SEC, * CCX SECONDARY DEFINITION
 TADDR=C1, * CCX SECONDARY DEFINITION
 TYPE=P *
*
 X25.PU PUTYPE=2, *
 VPACING=2, *
 MODETAB=LSPRITAB, * VTAM USER DEFINED TABLE
 XID=YES, * LEN SUPPORT
 DATMODE=HALF, *
 PACING=1, *
 SPAN=X2501, *
 ISTATUS=INACTIVE, *
 ANS=CONT *
*
 PLUPRI15 X25.LU LOCADDR=0, * INDEPENDANT LU

```

## Example B—IDNUM Values

```

DLOGMOD=LSPRI62B, * VTAM USER DEFINED TABLE *
ISTATUS=ACTIVE *
*

X25.VC LCN=5, *
VCCINDX=3, *
LLC=LLC4, *
MAXDATA=2000, *
VPACING=20, *
TYPE=P

*

* THIRD MCH FOR INN (REMOTE LOADING) ZL66
** RELATED WITH THE GEN NAMED NCPREM LOADED FROM DISK INTO REMOTE CCU

X25.MCH ADDRESS=66, *CONTROLLER LINE ADDR *
ANS=CONT, *
LCGDEF=(0,1), *MAX LCN IS 7 ON LCGN 0 *
FRMLGTH=131, *
MWINDOW=7, *LAP WINDOW *
STATION=DTE, *
TPTIMER=3, *LAPB RECOVERY PARAMETERS *
TDTIMER=1, *
NPRETRY=7, *
NDRETRY=1, *
LLCLIST=(LLC3), *
LCN0=USED, *
DBIT=NO, *NO DBIT SUPPORT ON THIS MCH *
ITRACE=YES, * LAP INTERNAL TRACE ACTIVE *
SPEED=9600, *REAL MCH SPEED *
TITIMER=1, *REAL T1 OF THE DCE *
SVCINN=1, *NB OF SIMULT.SWINN CONNECT. *
SDRTCNT=1, *NB OF RETRIES *
SHM=NO, *NO SHM FOR REMOTE *
SDRTIME=10 *TIMER BETWEEN RETRIES *
*
X25.LCG LCGN=0
*
***** (PVC) *****
*
X25.LINE LCN=0,VCCINDX=3,LLC=LLC3,TYPE=P, *
DSTNODE=INN
*
X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE
*
***** (SVC) *****
*
X25.LINE DSTNODE=INN, *MANDATORY FOR INN *
TYPE=S
*
X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
X25.VC LCN=1,VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1
*

* FOURTH MCH FOR GATE X.21 SWITCHED ACCESS ZL67

```

## Example B—IDNUM Values

```

*

X25.MCH ADDRESS=67, *
 ANS=CONT, *
 LCGDEF=(0,1), *
 FRMLGTH=131, *
 MWINDOW=5, LAP WINDOW *
 STATION=DTE, *
 TPTIMER=3, *
 TDTIMER=2, *
 NPRETRY=10, *
 NDRETRY=2, *
 LCN0=NOTUSED, *
 GATE=GENERAL, *
 SPAN=X2501, *
 LLCLIST=(LLC4), *
 X21SW=YES, *X21 SWITCH ACCESS TO ISDN NETWORK *
 X21NTWK=1984, *
 X21RTYTO=4.0, *X21 CALL OUT RETRY TIME *
 X21RTYCT=5, *X21 CALL OUT RETRY COUNT *
 X21INACT=40, *
 SPEED=64000 SPECIFIED TO SELECT LIC4B
*
X25.LCG LCGN=0
*
** 1 SVC'S *****
*
X25.VC LCN=1, *
 VCCINDX=3, *
 OUFINDX=1, *
 CALL=INOUT, *
 TYPE=SWITCHED
*

*
*
X25.END
* GENERATED BY X25NPSI
Z25P64A GROUP DIAL=NO, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALPBDT,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL64000 LINE IPL=NO, *
 SPAN=X2501, *
 UACB=XA64000
* GENERATED BY X25NPSI

```



```

ZC64000 SERVICE
* GENERATED BY X25NPSI
ZP64000 PU ADDR=01,
 PUDR=NO,
 PUTYPE=1,
 XID=NO,
 SPAN=X2501,
 MAXDATA=2000
* GENERATED BY X25NPSI
ZU64000 LU LOCADDR=0,
 SPAN=X2501,
 VPACING=20,
 PACING=1,
 LUDR=NO
* GENERATED BY X25NPSI
Z25P65A GROUP DIAL=NO,
 LSPRI=YES,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALPBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65000 LINE SPAN=X2501,
 UACB=XA65000
* GENERATED BY X25NPSI
ZC65000 SERVICE
* GENERATED BY X25NPSI
ZP65000 PU ADDR=01,
 ANS=CONT,
 DATMODE=HALF,
 PACING=1,
 ISTATUS=INACTIVE,
 SPAN=X2501,
 VPACING=2,
 PUTYPE=2,
 XID=YES,
 PUDR=NO
* GENERATED BY X25NPSI
SLUSEC26 LU LOCADDR=0,
 LUDR=NO
* GENERATED BY X25NPSI
Z25P65C GROUP DIAL=NO,
 LSPRI=YES,
 MODE=SEC,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,

```

## Example B—IDNUM Values

```
COMPTAD=NO, *
COMPOWN=YES, *
COMPSWP=NO, *
COMPACB=NO, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALNATER,,BALNATST,BALNATLS), *
TYPE=NCP, *
USERID=(5688035,BALPBDT,NORECMS), *
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65004 LINE SPAN=X2501, *
 TADDR=C1, *
 UACB=XA65004
* GENERATED BY X25NPSI
ZC65004 SERVICE
* GENERATED BY X25NPSI
ZP65004 PU ANS=CONT, *
 DATMODE=HALF, *
 PACING=1, *
 ISTATUS=INACTIVE, *
 MODETAB=LSPRITAB, *
 SPAN=X2501, *
 VPACING=2, *
 PUTYPE=2, *
 XID=YES, *
 PUDR=NO
* GENERATED BY X25NPSI
PLUPRI15 LU LOCADDR=0, *
 DLOGMOD=LSPRI62B, *
 ISTATUS=ACTIVE, *
 LUDR=NO
* GENERATED BY X25NPSI
Z25P65D GROUP DIAL=NO, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALPBDT,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65005 LINE IPL=NO, *
 UACB=XA65005
* GENERATED BY X25NPSI
ZC65005 SERVICE
* GENERATED BY X25NPSI
ZP65005 PU ADDR=01, *
 PUDR=NO, *
 PUTYPE=1, *
 XID=NO, *
```

```

MAXDATA=2000
* GENERATED BY X25NPSI
ZU65005 LU LOCADDR=0,
VPACING=20,
PACING=1,
LUDR=NO
* GENERATED BY X25NPSI
Z25P66A GROUP DIAL=NO,
LEVEL2=BALNAVL2,
LEVEL3=BALNAVL3,
LEVEL5=NCP,
LINEADD=NONE,
COMPTAD=NO,
COMPOWN=YES,
COMPSWP=NO,
COMPACB=NO,
LNCTL=SDLC,
NPACOLL=NO,
TIMER=(BALNATER,,BALNATST,BALNATLS),
TYPE=NCP,
USERID=(5688035,BALPBDT,NORECMS),
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL66000 LINE UACB=XA66000
* GENERATED BY X25NPSI
ZC66000 SERVICE
* GENERATED BY X25NPSI
ZP66000 PU ANS=CONT,
TGN=2,
ISTATUS=INACTIVE,
PUTYPE=4,
PUDR=NO
* GENERATED BY X25NPSI
Z25S67A GROUP DIAL=YES,
LEVEL2=BALNAVL2,
LEVEL3=BALNAVL3,
LEVEL5=NCP,
LINEADD=NONE,
COMPTAD=NO,
COMPOWN=YES,
COMPSWP=NO,
COMPACB=NO,
LINEAUT=YES,
LNCTL=SDLC,
NPACOLL=NO,
TIMER=(BALNATER,,BALNATST,BALNATLS),
TYPE=NCP,
USERID=(5688035,BALSBDT,NORECMS),
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL67001 LINE CALL=INOUT,
UACB=XA67001
* GENERATED BY X25NPSI
ZP67001 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
Z25S66B GROUP DIAL=YES,
LEVEL2=BALNAVL2,
LEVEL3=BALNAVL3,

```

## Example B—IDNUM Values

```

LEVEL5=NCP, *
LINEADD=NONE, *
COMPTAD=NO, *
COMPOWN=YES, *
COMPSWP=NO, *
COMPACB=NO, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALNATER,,BALNATST,BALNATLS), *
TYPE=NCP, *
USERID=(5688035,BALIBDT,NORECMS), *
PUTYPE=4, *
SHM=NO, *
ACTIVTO=0.0, *
BRKCON=NONE, *
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL66GGH LINE CALL=INOUT, *
UACB=XA66GGH
* GENERATED BY X25NPSI
ZP66GGH PU ISTATUS=INACTIVE, *
PUTYPE=4
* GENERATED BY X25NPSI
Z25S65B GROUP DIAL=YES, *
LSPRI=YES, *
LEVEL2=BALNAVL2, *
LEVEL3=BALNAVL3, *
LEVEL5=NCP, *
LINEADD=NONE, *
COMPTAD=NO, *
COMPOWN=YES, *
COMPSWP=NO, *
COMPACB=NO, *
LINEAUT=YES, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALNATER,,BALNATST,BALNATLS), *
TYPE=NCP, *
USERID=(5688035,BALSBDT,NORECMS), *
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
ZL65003 LINE CALL=INOUT, *
ROLE=PRI, *
ID=06344448, *
UACB=XA65003
* GENERATED BY X25NPSI
ZP65003 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
ZL65002 LINE CALL=INOUT, *
ROLE=PRI, *
ID=06344446, *
UACB=XA65002
* GENERATED BY X25NPSI
ZP65002 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
ZL65001 LINE CALL=INOUT, *
ROLE=PRI, *
ID=06344444, *

```

```

 UACB=XA65001
* GENERATED BY X25NPSI
ZP65001 PU PUTYPE=(1,2)
* GENERATED BY X25NPSI
Z25S64B GROUP DIAL=YES,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LINEAUT=YES,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALSBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
ZL64002 LINE CALL=INOUT,
 SPAN=X2501,
 UACB=XA64002
* GENERATED BY X25NPSI
ZP64002 PU SPAN=X2501,
 PUTYPE=(1,2)
* GENERATED BY X25NPSI
ZL64001 LINE CALL=INOUT,
 SPAN=X2501,
 UACB=XA64001
* GENERATED BY X25NPSI
ZP64001 PU SPAN=X2501,
 PUTYPE=(1,2)
* GENERATED BY X25NPSI
GRZ1 GROUP DIAL=NO,
 BERPROC=BALNUMBER,
 COMPACB=YES,
 COMPTAD=YES,
 COMPOWN=YES,
 COMPSWP=YES,
 LEVEL2=BALNAML2,
 LEVEL3=BALNAML3,
 LEVEL5=NCP,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALLAP4,,BALLAP4,BALLAP4),
 TYPE=NCP,
 USERID=(5688035,BALMBDT,NORECMS,P),
 XIO=(BALNAMXL,BALNAMXS,BALNAMXI,BALNAMXX)
* GENERATED BY X25NPSI
ZL64 LINE ADDRESS=(64,FULL),
 SPAN=X2501,
 XMONLNK=YES,
 UACB=(X25A64X,X25A64R)
* GENERATED BY X25NPSI
ZC64 SERVICE

```

## Example B—IDNUM Values

```
* GENERATED BY X25NPSI
ZP64 PU ADDR=01,
 ANS=CONT,
 SPAN=X2501,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
*
* GENERATED BY X25NPSI
ZU64 LU LOCADDR=0,
 SPAN=X2501,
 VPACING=(2,1),
 ISTATUS=ACTIVE
*
* GENERATED BY X25NPSI
GRZ2 GROUP DIAL=NO,
 BERPROC=BALNUMBER,
 COMPACB=YES,
 COMPTAD=YES,
 COMPOWN=YES,
 COMPSWP=YES,
 LEVEL2=BALNAML2,
 LEVEL3=BALNAML3,
 LEVEL5=NCP,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALLAP4,,BALLAP4,BALLAP4),
 TYPE=NCP,
 USERID=(5688035,BALNBDT,NORECMS,P),
 XIO=(BALNAMXL,BALNAMXS,BALNAMXI,BALNAMXX)
*
* GENERATED BY X25NPSI
ZL65 LINE ADDRESS=(65,FULL),
 XMONLNK=NO,
 UACB=(X25A65X,X25A65R)
*
* GENERATED BY X25NPSI
ZC65 SERVICE
* GENERATED BY X25NPSI
ZP65 PU ADDR=01,
 ANS=CONT,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
*
* GENERATED BY X25NPSI
ZU65 LU LOCADDR=0,
 VPACING=(2,1),
 ISTATUS=ACTIVE
*
* GENERATED BY X25NPSI
ZU65L1 LU LOCADDR=1,
 VPACING=(2,1),
 ISTATUS=ACTIVE
*
* GENERATED BY X25NPSI
ZL66 LINE ADDRESS=(66,FULL),
 XMONLNK=NO,
 UACB=(X25A66X,X25A66R)
*
* GENERATED BY X25NPSI
ZC66 SERVICE
* GENERATED BY X25NPSI
ZP66 PU ADDR=01,
```

## Example B—IDNUM Values

```

 ANS=CONT,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
* GENERATED BY X25NPSI
ZU66 LU LOCADDR=0,
 VPACING=(2,1),
 ISTATUS=INACTIVE
* GENERATED BY X25NPSI
ZL67 LINE ADDRESS=(67,FULL),
 SPAN=X2501,
 SPEED=55855,
 XMONLNK=NO,
 UACB=(X25A67X,X25A67R)
* GENERATED BY X25NPSI
ZC67 SERVICE
* GENERATED BY X25NPSI
ZP67 PU ADDR=01,
 ANS=CONT,
 SPAN=X2501,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
* GENERATED BY X25NPSI
ZU67 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
*

CZGROUP GROUP LNCTL=CA,
 CA=TYPE6,
 DELAY=0.0,
 NCPCA=ACTIVE
CZLINE5 LINE ADDRESS=00
CZ5PU PU PUTYPE=5
CZLINE1 LINE ADDRESS=08
CZ1PU PU PUTYPE=5

 GENEND INIT=BALINIMD,
 TMRTICK=BALTICK,
 UGLOBAL=BALNMGOP

```

Switched Major Node—Example B2

```

/** THIS JCL UPDATE THE SYS1.VTAMLST (MVS6)
/** GET THE FILE TO BE UPDATED UNDER THE ./ ADD CHANGE ITS NAME.
//UPD EXEC PGM=IEBUPDTE,PARM=NEW
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD DISP=SHR,DSN=SYS1.VTAMLST,VOL=SER=MVSVM3,UNIT=3350
//SYSIN DD DATA
// ADD NAME=PT3346MN MEMBER NAME

* SWITCHED MAJOR NODE TO BE USED IN CORRELATION WITH PT3346Z GEN

**
PT3346MN VBUILD MAXGRP=20, *
 MAXNO=20, *
 TYPE=SWNET
**

*
* ENTRIES FOR GATE *
* FOR GATE, CALLIN IS ALWAYS SIMULATED TO VTAM. PATH NOT NEEDED *
*
* TO USE WITH GEN:PT3346Z, MCH:ZL64-ZP64-ZU64 *
* SVC:ZL64001 DEFAULT IDNUM:06220006 (CALLIN GATEN <<<<<I *
* SVC:ZL64002 DEFAULT IDNUM:06220008 (CALLOUT GATEN-----I *
*

*
GATEPI PU ADDR=01, GATE INCALL *
 IDBLK=062, 1ST GATE VC (INCOMING CALL) *
 IDNUM=20006, 3TH SVC FROM LAST SVC (3*2)= 6 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
GATELI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
*
**
GATEPO PU ADDR=01, 2ND GATE VC USED(CALL OUT WITHOUT PATH) *
 IDBLK=062, *
 IDNUM=20008, 4TH SVC FROM LAST (4*2)=8 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
GATELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES

*
* ENTRIES FOR X21 GATE *
* FOR GATE, CALLIN IS ALWAYS SIMULATED TO VTAM. PATH NOT NEEDED *
* SVC:ZL67001 DEFAULT IDNUM:00320002 *

```



## Example B—IDNUM Values

```

*

*
BATEPO PU ADDR=01, 2ND GATE VC USED(IN OR OUT CALL) *
 IDBLK=003, DEFAULT VALUE (IDBLKG NOT CODED IN GEN) *
 IDNUM=20002, 1ST SVC FROM LAST SVC (1*2)=2 *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
BATELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES *
*

* ENTRIES FOR PAD *
*
* SVC:ZL64001 DEFAULT IDNUM:06320002 (CALLIN PAD) *
* SVC:ZL64002 DEFAULT IDNUM:06320008 (CALLIN CALLOUT PAD) *
*

*
PADIPI PU ADDR=01, PADI INCALL OR 1ST GATE VC USED *
 IDBLK=063, (IN OR OUT CALL) *
 IDNUM=20008, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 MAXPATH=1, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PADILI LU LOCADDR=0, 0 IS REQUIRED FOR NON SNA X25 DEVICES *
 SSCPFM=USSNTO, *LOGON ENDS WITH CRLF *
 MODETAB=MODRAL, *MODETAB FOR LU1 3767/TWX *
 DLOGMOD=SSCICSF, *FLIPFLOP OR CONTENTION PROT *
 TERM=TWX, *MUST BE DEFINED FOR TWX/3101 *
 USSTAB=USSTWX *SPECIAL USSTAB FOR TWX/3101 *
*
*
PADIPO PU ADDR=01, PADI INCALL OR 1ST GATE VC USED *
 IDBLK=063, (IN OR OUT CALL) *
 IDNUM=20006, NON SNA CALL OUT. *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=2000, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 MAXPATH=1, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
*
PADPATH PATH DIALNO=10605023650201*20006, L=5 PAD (MODEM PAD) *
 GRPNM=Z25S64B,GID=2,PID=31 Z25S64B=GROUP OF SWITCHED

```

## Example B—IDNUM Values

```

PADILO LU LOCADDR=0, *FOR NON SNA X25 DEVICES *
 SSCPFM=USSNTO, *LOGON ENDS WITH CRLF *
 MODETAB=MODRAL, *MODETAB FOR LU1 3767/TWX *
 DLOGMOD=SSCICSF, *FLIPFLOP OR CONTENTION PROT *
 TERM=TWX, *MUST BE DEFINED FOR TWX/3101 *
 USSTAB=USSTWX *SPECIAL USSTAB FOR TWX/3101 *

* ENTRIES FOR PCNE
* SVC:ZL64001 IDNUM=ZZZZ OF CUD FIELD *
* SVC:ZL64002 *

*
PCNEPI PU ADDR=01, PCNE TO PCNE INBOUND *
 IDBLK=064, *
 IDNUM=22225, IDNUM=ZZZZ OF INCOM CALL CUD FIELD *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXOUT=1, TO LET LOGON MSG IN AFTER ACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNELI LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
**
PCNEPO PU ADDR=01, PCNE TO PCNE OUTBOUND *
 IDBLK=064, *
 IDNUM=22224, EQUAL ZZZZ *
 MAXPATH=6, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=1024, SEGMENTING NOT SUPPORTED *
 PACING=1, PACING=1 IS REQUIRED *
 VPACING=2, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 PUTYPE=1, *
 SSCPFM=USSSCS
PCNELO LU LOCADDR=0 0 IS REQUIRED FOR NON SNA X25 DEVICES
**
PCNEP1 PATH DIALNO=10604016500201*22224, ZZZZ+1=CUD FIELD OF CALLRQ*
 GRPNM=Z25S64B,GID=2,PID=20 Z25S64B=GROUP OF SWITCHED
 VC'S ON INVOLVED MCH IN GENERATION
*

* ENTRY FOR INCOMING AND OUTGOING CALLS FOR BNN PSH AND QLLC*
* WITH CANNED XID. GEN=PT3346Z MCH=65 (ZL65 ZP65) *
* ID=0634444 *
* 6 *
* 8 *

PSHPO PU ADDR=01, *
 IDBLK=063, *
 IDNUM=41264, IDNUM OF REMOTE PU *
 MAXPATH=3, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=265, SEGMENTING SUPPORTED *
 PACING=3, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 VPACING=3, *
 PUTYPE=2, *

```

## Example B—IDNUM Values

```

SSCPFM=USSSCS OUTGOING CALL FOR PSH
PSHP PATH DIALNO=10604016420201, L=2 FOR PSH MODEM PTT1*
 GRPNM=Z25S65B,GID=2,PID=31 Z25S65B=GROUP OF SWITCHED
*
 VC'S ON INVOLVED MCH IN GENERATION PT3346Z
PSHLO LU LOCADDR=2,MODETAB=MODTD3
*
PSHPI PU ADDR=01, *
 IDBLK=063, *
 IDNUM=44444, IDNUM FOR ANONYMOUS CALLING PU *
 MAXPATH=3, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=265, SEGMENTING SUPPORTED *
 PACING=3, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 VPACING=3, *
 PUTYPE=2, *
 SSCPFM=USSSCS
PSHLI LU LOCADDR=2,MODETAB=MODTD3
*
QBNNPO PU ADDR=01, *
 IDBLK=063, *
 IDNUM=44446, IDNUM FOR ANONYMOUS CALLING PU *
 MAXPATH=3, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=265, SEGMENTING SUPPORTED *
 PACING=3, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 VPACING=3, *
 PUTYPE=2, *
 SSCPFM=USSSCS
QBNNP PATH DIALNO=10604016530101, L=3 FOR BNN QLLC MODEM PTT2*
 GRPNM=Z25S65B,GID=2,PID=32 Z25S65B=GROUP OF SWITCHED
*
 VC'S ON INVOLVED MCH IN GENERATION PT3346Z
QBNNLO LU LOCADDR=2,MODETAB=MODTD3
PSHPI2 PU ADDR=01, *
 IDBLK=063, *
 IDNUM=44448, IDNUM FOR ANONYMOUS CALLING PU *
 MAXPATH=3, *
 DISCNT=YES, DISCONTACT/ABCONN ON LAST DACTLU *
 MAXDATA=265, SEGMENTING SUPPORTED *
 PACING=3, *
 ANS=CONT, FOR SW SESSION CONTINUATION *
 VPACING=3, *
 PUTYPE=2, *
 SSCPFM=USSSCS
PSHLI2 LU LOCADDR=2,MODETAB=MODTD3

* ENTRY FOR SWITCHED CASUAL CONNECTION *
* - IFNYL7P IS A CEN(VTAM/NCP) WORKING AS A NODE T2.1* *

*
IFNYL7P PU ADDR=01, *
 MAXPATH=1, *
 DISCNT=YES, *
 PACING=1, *
 VPACING=2, *
 PUTYPE=2, *
 ANS=CONT, *

```

## Example B—IDNUM Values

```

 ROLE=PRI, PRIMARY ROLE FORCED DURING ROLE NEGOTIATION *
 MODETAB=LSPRITAB, VTAM USER DEFINED TABLE *
 SSCPFM=USSSCS, *
 CPNAME=IFNYL7 ADJACENT LINK STATION SSCP NAME
 (REPLACES IDNUM/IDBLK FOR PUT2.1)
*
**
IFNYL7P1 PATH DIALNO=10604016930201, (ALS MODEM) *
 GRPNM=Z25S65B, *
 GID=2, *
 PID=33
*
SLUSEC26 LU LOCADDR=0, IDEPENDANT LU DEFINITION (NEEDED FOR CCX) *
 DLOGMOD=LSPRI62B VTAM USER DEFINED TABLE

* ENTRY FOR REMOTE LOAD

* VTAM....NCP SA46-----| X |-----NCP SA45
* | 25 |
*

* SWITCHED MAJOR NODE FOR X25 FOR BOTH NCP'S WHICH ARE: *
* - SA46 LOCAL *
* - SA45 REMOTE (RELATED REMOTE GEN IS REMNCP) *
* *

**
** FOR OUTGOING CALL ISSUED FROM LOCAL TO REMOTE
*
SADREM PU PUTYPE=4, * LOADABLE REMOTE PU *
 ADDR=01, * VALUE ANY (OPERAND IGNORED FOR SAD CONN.) *
 TGN=3, *
 SUBAREA=45, *
 IDNUM=AAAAA, *
 IDBLK=003, *
 NETID=FRIBM412, *
 ANS=CONT, *
 MAXPATH=1, *
 MAXDATA=1024
*
LOCPH PATH DIALNO=10604016410201, L=1 FOR INN SWITCHED MODEM PTT1*
 GRPNM=Z25S66B,PID=34
*
SADLOC PU PUTYPE=4, * PU LOADER *
 ADDR=01, * VALUE ANY (OPERAND IGNORED FOR SAD CONN.) *
 TGN=3, *
 SUBAREA=46, *
 IDNUM=AAAAA, *
 IDBLK=003, *
 NETID=FRIBM412, *
 ANS=CONT, *
 MAXPATH=1, *
 MAXDATA=1024

```

### NCP and NPSI Connection Input—Example B3

Example B3 is of a remote NCP that contains a link to example B1 upon which this NCP can be loaded.

```
* EXAMPLE B3

* GENERATION NAME: REMNCPG
*
* MCH 01 (CL1) PVC0 FOR REMOTE LOAD
* SVC1 FOR REMOTE LOAD
*
* REMNCPG IS A MINI NCP RECORDED ON CCU DISK OR DISKETTE
*

*
* OPTIONS USERGEN=X25NPSI, *INDICATE NEO NAME X
* NEWDEFN=(YES,ECHO) *INCLUDE NPSI STMTS IN VTAMLST
ATTACHED PCCU DUMPDS=DUMPX25,SUBAREA=16,VFYL=YES
*
BAKBUILD BUILD NPA=YES, X
 VERSION=V5R3, * NCP VERSION FOR 3745 X
 PUNAME=MANETREM, * NCP NAME X
 LOADLIB=LNCP10, X
 TYPGEN=NCP, X
 MODEL=3745, X
 USGTIER=4, X
 X25.USGTIER=4, X
 NETID=FRIBM412, * HOST ID X
 MAXSSCP=8, * MAX SSCP OWNER X
 MAXSUBA=63, X
 SUBAREA=45, * SAD CALLER X
 BFRS=124, * BEST FOR NPSI PERFORMANCE X
 LTRACE=4, * ALLOWS LINE TRACE ON TWO MCH'S X
 OLT=YES, X
 DR3270=NO, X
 BRANCH=8000, X
 TRACE=(YES,100), * 100 ENTRIES IN ADDRESS TRACE TBL X
 NEWNAME=REMNCPG, * SAD CALLER X
 X25.PREFIX=C, * ALL NAMES START WITH C X
 X25.IDNUMH=2, * X25.IDNUMH MUST MATCH WITH SWITCHX
 TYP SYS=OS, * MAJOR NODE IDS.(2FIRST DIGITS) X
 X25.SNAP=YES, * OPTIONAL DIAGNOSIS SNAP INCLUDED X
 X25.MCHCNT=1, * NUMBER OF PHYSICAL LINKS X
 X25.MAXPIU=64K
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD,
 RIMM,NAKLIM,SESSION,SSPAUSE,XMTLMT)
X
*
SMMFTAB1 SWMMT IDNUM=AAAAA, X
 NETID=FRIBM412, X
 MAXOUT=7
*
MVS6 HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16,
 DELAY=0.1,TIMEOUT=30
X
*
PATH16 PATH DESTSA=(16),ER0=(16,1),ER1=(46,2),ER2=(46,3)
PATH46 PATH DESTSA=(46),ER1=(46,2),ER2=(46,3)
*
```

## Example B—IDNUM Values

```

 LUDRPOOL NUMTYP1=10, * FOR NON SNA CALLS X
 NUMTYP2=10 * FOR QLLC OR PSH CALLS

 EJECT * AT LEAST ONE SDLC LINE
CDLC6 GROUP LNCTL=SDLC, * MUST BE DEFINED X
 TYPE=NCP,
 DIAL=YES,
 REPLYTO=20

CL6 LINE ADDRESS=6, *
 CLOCKNG=EXT, *
 NEWSYNC=NO,
 DUPLEX=HALF,
 NRZI=NO,
 CALL=INOUT,
 RETRIES=(7,0,0)
CP6 PU ISTATUS=INACTIVE,
 PUTYPE=2

CGRNPA GROUP LNCTL=SDLC, * FOR NPM RECORDING X
 TYPE=NCP,
 DIAL=NO,
 DISCNT=NO,
 VIRTUAL=YES,
 NPARSC=YES,
 ISTATUS=INACTIVE
CLNNPA LINE LINEFVT=NPAVFVT
CPUNPA PU PUTYPE=2,PUFVT=NPAPFVT
CLUNPA LU MAXCOLL=6,LUFVT=(NPALFVT,NPALFVT),
 VPACING=0,LOCADDR=1

 EJECT
***** X25 NPSI INPUT
* ALLOW 3 VCCPT AND OUFT ENTRIES
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES,RESETINO=NO
*
 X25.VCCPT INDEX=1,
 MAXPKTL=128,
 VWINDOW=1, * PACKET LEVEL WINDOW
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.VCCPT INDEX=2,
 MAXPKTL=128,
 VWINDOW=2,
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.VCCPT INDEX=3,
 MAXPKTL=128,
 VWINDOW=3,
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.OUFT INDEX=1
*
 X25.OUFT INDEX=2, * PKT SIZE=128,VWINDOW=2
 OPTFACL=420707430202
*
 X25.OUFT INDEX=3, * PKT SIZE=128,VWINDOW=3, AND
 OPTFACL=03430102420707430303, * REVERSECHARGING REQUEST X

```

```

USRFILD=1234567890

*
 EJECT

* FIRST MCH CL1: FOR SUBAREA DIAL

 X25.MCH ADDRESS=1, *CONTROLLER LINE ADDR X
 ANS=CONT, X
 LCGDEF=(0,1), X
 FRMLGTH=131, X
 MWINDOW=7, *LAP WINDOW X
 STATION=DTE, X
 XMONLNK=YES, X
 TPTIMER=3, *LAPB RECOVERY PARAMETERS X
 TDTIMER=1, * X
 NPRETRY=7, * X
 NDRETRY=1, * X
 LLCLIST=(LLC3), X
 LCN0=USED, X
 SPAN=X2501, *FOR NETVIEW X
 ITRACE=YES, * LAP INTERNAL TRACE ACTIVE X
 SPEED=9600, *REAL MCH SPEED X
 T1TIMER=1, *REAL T1 OF THE DCE X
 SVCINN=1, *NB OF SIMULT.SWINN CONNECT. X
 SDRTCNT=1, *NB OF RETRIES X
 SDRTIME=10 *TIMER BETWEEN RETRIES X
*
 X25.LCG LCGN=0
*
***** (PVC) *****
*
CL100 X25.LINE LCN=0,VCCINDX=3,LLC=LLC3,TYPE=P,SPAN=X2501, X
 DSTNODE=INN,MONLINK=YES, X
 IPL=YES
*
CP100 X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE
*
***** (SVC) *****
*
*
CL102 X25.LINE DSTNODE=INN, *MANDATORY FOR INN X
 TYPE=S, X
 MONLINK=YES, X
 IPL=YES, X
 SWMMTID=SMMFTAB1 *
*
CP102 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
 X25.VC LCN=(1),VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1, X
 SPAN=X2501
*
 EJECT

 X25.END

CHAGRP GROUP LNCTL=CA, X
 CA=TYPE6, X

```

## Example B—IDNUM Values

```

 DELAY=0.0

CHALINE LINE NCPA=ACTIVE, X
 ADDRESS=8

CHAPU PU PUTYPE=5

 GENEND
 END

```



NCP Input Produced by NDF (NPSI) - Example B3

```

OPTIONS USERGEN=X25NPSI, *INDICATE NEO NAME *
 NEWDEFN=(YES,ECHO) *INCLUDE NPSI STMTS IN VTAMLST
ATTACHED PCCU DUMPDS=DUMPX25,SUBAREA=16,VFYLM=YES
*
BAKBUILD BUILD NPA=YES, *
 VERSION=V5R3, * NCP VERSION FOR 3745 *
 PUNAME=MANETREM, * NCP NAME *
 LOADLIB=LNCP10, *
 TYPGEN=NCP, *
 MODEL=3745, *
 USGTIER=4, *
 X25.USGTIER=4, *
 NETID=FRIBM412, * HOST ID *
 MAXSSCP=8, * MAX SSCP OWNER *
 MAXSUBA=63, *
 SUBAREA=45, * SAD CALLER *
 BFRS=124, * BEST FOR NPSI PERFORMANCE *
 LTRACE=4, * ALLOWS LINE TRACE ON TWO MCH'S *
 OLT=YES, *
 DR3270=NO, *
 BRANCH=8000, *
 TRACE=(YES,100), * 100 ENTRIES IN ADDRESS TRACE TBL *
 NEWNAME=REMNCPPG, * SAD CALLER *
 X25.PREFIX=C, * ALL NAMES START WITH C *
 X25.IDNUMH=2, * X25.IDNUMH MUST MATCH WITH SWITCH*
 TYP SYS=OS, * MAJOR NODE IDS.(2FIRST DIGITS) *
 X25.SNAP=YES, * OPTIONAL DIAGNOSIS SNAP INCLUDED *
 X25.MCHCNT=1, * NUMBER OF PHYSICAL LINKS *
 X25.MAXPIU=64K
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD,RIMM,NAKLIM*,
 ,SESSION,SSPAUSE,XMTLMT)
*
SMMFTAB1 SWMMT IDNUM=AAAAA, *
 NETID=FRIBM412, *
 MAXOUT=7
*
MVS6 HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=16,
 DELAY=0.1,TIMEOUT=30
*
PATH16 PATH DESTSA=(16),ER0=(16,1),ER1=(46,2),ER2=(46,3)
PATH46 PATH DESTSA=(46),ER1=(46,2),ER2=(46,3)
*
 LUDRPOOL NUMTYP1=10, * FOR NON SNA CALLS *
 NUMTYP2=10 * FOR QLLC OR PSH CALLS *

CDLC6 GROUP LNCTL=SDLC, * MUST BE DEFINED *
 TYPE=NCP, *
 DIAL=YES, *
 REPLYTO=20

CL6 LINE ADDRESS=6, *
 CLOCKNG=EXT, *
 NEWSYNC=NO, *
 DUPLEX=HALF, *
 NRZI=NO, *
 CALL=INOUT, *

```

## Example B—IDNUM Values

```

 RETRIES=(7,0,0)
CP6 PU ISTATUS=INACTIVE,
 PUTYPE=2

CGRNPA GROUP LNCTL=SDLC, * FOR NPM RECORDING
 TYPE=NCP,
 DIAL=NO,
 DISCNT=NO,
 VIRTUAL=YES,
 NPARSC=YES,
 ISTATUS=INACTIVE
CLNNPA LINE LINEFVT=NPAVFVT
CPUNPA PU PUTYPE=2,PUFVT=NPAPFVT
CLUNPA LU MAXCOLL=6,LUFVT=(NPALFVT,NPALFVT),
 VPACING=0,LOCADDR=1

***** X25 NPSI INPUT
* ALLOW 3 VCCPT AND OUFT ENTRIES
TRANSPAC X25.NET CPHINDX=3,OUHINDX=3,DM=YES,RESETINO=NO,
 DONE=YES
*
 X25.VCCPT INDEX=1,
 MAXPKTL=128,
 VWINDOW=1, * PACKET LEVEL WINDOW
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.VCCPT INDEX=2,
 MAXPKTL=128,
 VWINDOW=2,
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.VCCPT INDEX=3,
 MAXPKTL=128,
 VWINDOW=3,
 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
 X25.OUFT INDEX=1
*
 X25.OUFT INDEX=2, * PKT SIZE=128,VWINDOW=2
 OPTFACL=420707430202
*
 X25.OUFT INDEX=3, * PKT SIZE=128,VWINDOW=3, AND
 OPTFACL=03430102420707430303, * REVERSECHARGING REQUEST
 USRFILD=1234567890

*

* FIRST MCH CL1: FOR SUBAREA DIAL

 X25.MCH ADDRESS=1, *CONTROLLER LINE ADDR
 ANS=CONT,
 LCGDEF=(0,1), *MAX LCN IS 7 ON LCGN 0
 FRMLGTH=131,
 MWINDOW=7, *LAP WINDOW
 STATION=DTE,
 XMONLNK=YES,
 TPTIMER=3, *LAPB RECOVERY PARAMETERS
 TDTIMER=1, *

```

## Example B—IDNUM Values

```

NPRETRY=7, *
NDRETRY=1, *
LLCLIST=(LLC3), *
LCN0=USED, *
SPAN=X2501, *FOR NETVIEW
ITRACE=YES, * LAP INTERNAL TRACE ACTIVE
SPEED=9600, *REAL MCH SPEED
T1TIMER=1, *REAL T1 OF THE DCE
SVCINN=1, *NB OF SIMULT.SWINN CONNECT.
SDRTCNT=1, *NB OF RETRIES
SDRTIME=10 *TIMER BETWEEN RETRIES
*
X25.LCG LCGN=0
*
***** (PVC) *****
*
CL100 X25.LINE LCN=0,VCCINDX=3,LLC=LLC3,TYPE=P,SPAN=X2501,
 DSTNODE=INN,MONLINK=YES,
 IPL=YES
*
CP100 X25.PU PUTYPE=4,TGN=2,ANS=CONT,PUDR=NO,ISTATUS=INACTIVE
*
***** (SVC) *****
*
CL102 X25.LINE DSTNODE=INN, *MANDATORY FOR INN
 TYPE=S,
 MONLINK=YES,
 IPL=YES,
 SWMTID=SMMFTAB1 *
*
CP102 X25.PU PUTYPE=4,ISTATUS=INACTIVE
*
X25.VC LCN=(1),VCCINDX=3,TYPE=S,CALL=INOUT,OUFINDX=1,
 SPAN=X2501
*

X25.END
* GENERATED BY X25NPSI
C25P1A GROUP DIAL=NO,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALPBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
* GENERATED BY X25NPSI
CL100 LINE IPL=YES,
 SPAN=X2501,
 MONLINK=YES,

```

## Example B—IDNUM Values

```

 UACB=XA1000
* GENERATED BY X25NPSI
CC1000 SERVICE
* GENERATED BY X25NPSI
CP100 PU ANS=CONT,
 TGN=2,
 ISTATUS=INACTIVE,
 PUTYPE=4,
 PUDR=NO
* GENERATED BY X25NPSI
C25S1B GROUP DIAL=YES,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALIBDT,NORECMS),
 PUTYPE=4,
 SHM=YES,
 ACTIVTO=0.0,
 BRKCON=NONE,
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
* GENERATED BY X25NPSI
CL102 LINE IPL=YES,
 SWMTID=SMMFTAB1,
 CALL=INOUT,
 MONLINK=YES,
 UACB=XA1GGH
* GENERATED BY X25NPSI
CP102 PU ISTATUS=INACTIVE,
 PUTYPE=4
* GENERATED BY X25NPSI
CNET11 GROUP DIAL=NO,
 BERPROC=BALNUMBER,
 COMPACB=YES,
 COMPTAD=YES,
 COMPOWN=YES,
 COMPSWP=YES,
 LEVEL2=BALNAML2,
 LEVEL3=BALNAML3,
 LEVEL5=NCP,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALLAP4,,BALLAP4,BALLAP4),
 TYPE=NCP,
 USERID=(5688035,BALMBDT,NORECMS,P),
 XIO=(BALNAMXL,BALNAMXS,BALNAMXI,BALNAMXX)
* GENERATED BY X25NPSI
CL1 LINE ADDRESS=(1,FULL),
 SPAN=X2501,
 XMONLNK=YES,

```

```

 UACB=(X25A1X,X25A1R)
* GENERATED BY X25NPSI
CC1 SERVICE
* GENERATED BY X25NPSI
CP1 PU ADDR=01,
 ANS=CONT,
 SPAN=X2501,
 MAXDATA=261,
 PUDR=NO,
 XID=NO,
 PUTYPE=1
* GENERATED BY X25NPSI
CU1 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=INACTIVE

CHAGRP GROUP LNCTL=CA,
 CA=TYPE6,
 DELAY=0.0

CHALINE LINE NCPA=ACTIVE,
 ADDRESS=8

CHAPU PU PUTYPE=5

 GENEND INIT=BALINIMD,
 TMRTICK=BALTICK,
 UGLOBAL=BALNMGOP

```

## Example C—NPSI NPM and FTPI (V3R4)

This section contains one example that shows NCP and NPSI generation input using NPM and FTPI.

### NCP and NPSI Generation Input with NPM and FTPI—EXAMPLE C

```

* *****

* *****
* * SA=21 * * SA=44 * DCE DTE * SA=42 * * SA=21 **
* * HOST * *PUNAME: * *PUNAME: * * HOST **
* * LGETMXA5 * * MANETLOC * *DEGASLOC * *LGETMXA5**
* * PATH= * * * * * * * * PATH= **
* * PAS21PER * *LOADMODULE* *LOADMODULE:* *PAS21PER**
* * * * * * * * * * * * * *
* *****
*

* GENERATION OF THE X25 BLOCKS FOR NEO WITHIN NCP *
* X.25 NPSI MCH DESCRIPTION *

* L *
* G L *
* C C V *
* LINE N N C *

* - MCH1 (032) (032) LINE ADDRESS V3R4*
* 0-00 NOT USED V3R4*
* V35-56000Bps-DTE V3R4*
* 1-00 1-80 PVC FTPI 3 PVC V3R4*
* 1-81 1-FF SVC FTPI 3 SVC IN/OUT V3R4*

* TO BE LOADED ON 3745-170 CUA=0D0 or 0C0 *
* *
* *

EJECT

*
* GENERATION OF THE X25 BLOCKS FOR NEO WITHIN NCP *
* *

*
OPTIONS NEWDEFN=(YES,ECHO),USERGEN=(X25NPSI,FNMNDFGN)
*
IFNY21 PCCU AUTODMP=NO, X
AUTOIPL=NO, X
AUTOSYN=YES, X
DUMPDS=DUMPX25, X
MAXDATA=8640, X
SUBAREA=21, X
VFYLM=YES, X
NETID=FRIBM412
*
IFNY13 PCCU AUTODMP=NO, X

```

```

AUTOIPL=NO, X
AUTOSYN=YES, X
DUMPDS=DUMPX25, X
MAXDATA=8640, X
SUBAREA=13, X
VFYLM=YES, X
NETID=FRIBM412
*

*

PT3442A BUILD MAXSUBA=63, B
MODEL=3745-170, B
SUBAREA=42, B
TYPGEN=NCP, B
BACKUP=5, B
TYP SYS=MVS, B
BFRS=72, B
BRANCH=800, B
CATRACE=(YES,255), B
DSABLTO=6.5, B
ENABLTO=6.5, B
LOADLIB=NCPLD, B
LTRACE=2, B
MAXSSCP=4, B
NETID=FRIBM412, B
NEWNAME=PT3442A, B
NUMHSAS=5, FOR EXPANSION B
NPA=YES, B
OLT=NO, B
PUNAME=DEGASLOC, B
SLODOWN=12, B
SESSACC=(YES,ALL), B
TRANSFR=120, B
USGTIER=4, B
VERSION=V5R4, B
VRPOOL=12, B
X25.PREFIX=D, B
X25.IDNUMH=2, B
X25.BYTHROD=65536, B
X25.SEGTHROD=65536, B
X25.SNAP=YES, B
X25.MCHCNT=1, B
X25.MAXPIU=8000, B
X25.USGTIER=4
*
*
SYSCNTRL OPTIONS=(S
BHSASSC, S
ENDCALL, S
MODE, S
NAKLIM, S
RCOND, S
RECMD, S
RIMM, S
SESSION, S
SSPAUSE, S
STORDSP, S

```

# Example C (NPSI Only)

```

XMTLMT)
*
*
BAKMXA5 HOST MAXBFRU=51, X
 INBFRS=3, X
 UNITSZ=172, X
 BFRPAD=(0), X
 SUBAREA=(21)
*
*
BAKMSP1 HOST MAXBFRU=51, X
 INBFRS=3, X
 UNITSZ=172, X
 BFRPAD=(0), X
 SUBAREA=(13)
*
*
*
 PATH DESTSA=(13), X
 ER3=(13,1),ER6=(13,1),ER7=(13,1),
 VR3=3,VR6=6,VR7=7 X
*
 PATH DESTSA=(21), X
 ER3=(21,1),ER6=(21,1),ER7=(21,1),
 VR3=3,VR6=6,VR7=7 X
*
*
 PUDRPOOL NUMBER=5
 LUDRPOOL NUMTYP1=30,
 NUMTYP2=100,
 NUMILU=500

* SDLC STATIONS *

* SDLC LEASED 3276 LU TYPE 2 SDLC LINE 11 *

XGP3276 GROUP LNCTL=SDLC,
 TYPE=NCP,
 DIAL=NO,
 REPLYTO=3
XL13276 LINE ADDRESS=075,CLOCKNG=EXT,DUPLEX=FULL,SPEED=2400, NEW
 PUDR=YES,MAXPU=4,NRZI=NO,LPDATS=LPDA1
 SERVICE ORDER=(XP13276),MAXLIST=14
XP13276 PU ADDR=01, PU: NAME MUST MATCH TCAM
 MAXDATA=261, CONTROL INFOR PLUS 256 DATA
 MAXOUT=7,
 PUDR=YES, DR INCLUDED PU MAY BE DELETED
 PUTYPE=2,
 PASSLIM=9 ALLOW FULL SCREEN WRITE (9X256)
XU13276 LU LOCADDR=2,PACING=(1) 3277 LU: NAME MUST MATCH TCAM
XU23276 LU LOCADDR=3,PACING=(1) 3277 LU: NAME MUST MATCH TCAM
XU33276 LU LOCADDR=4,PACING=(1) 3277 LU: PRINTER """"""""""
*
 EJECT
GRNPM D GROUP LNCTL=SDLC, * FOR NPM RECORDING X
 TYPE=NCP, X
 DIAL=NO, X

```



```

DISCNT=NO, X
VIRTUAL=YES, X
NPARSC=YES, X
ISTATUS=ACTIVE

LNNPMD LINE
PUNPMD PU MODETAB=X25MOTAB
LUNPMD1 LU DLOGMOD=LU2V3P2
LUNPMD2 LU DLOGMOD=LU2V6P2
LUNPMD3 LU DLOGMOD=LU2V7P2

***** X25 NPSI INPUT

* CTCP DEFINITION FOR FTPI

X25.CPL CTCPCNO=0,MINDATA=02,MAXTIME=1,ALTCTCP=1, X
PUNAME=PUDFTPI0, X
LUNAME=LUDFTPI0, X
LOGAPPL=CFTPI, X
ISTATUS=ACTIVE

*
X25.CPL CTCPCNO=1,MINDATA=02,MAXTIME=1,ALTCTCP=0, X
PUNAME=PUDFTPI1, X
LUNAME=LUDFTPI1, X
LOGAPPL=EFTPI, X
ISTATUS=ACTIVE

*
*

* NETWORK DEFINITION *

* ALLOW 3 VCCPT AND OUFT ENTRIES
XIPAC X25.NET CPHINDX=4, * NB OF VCCPT ENTRIES X
OUHINDX=4, * NB OF OUFT ENTRIES X
NSTDFAC=(49), * TRANSIT DELAY SEL. NOT REJ SJMV3R3
DM=YES, * LAPB DM IS SUPPORTED X
RESETINO=(0085,0086,8183,8184,8185,8186,8188,8189,8987,XC
X82) * THESE RESET CC DD WILL NOT INOP

*
X25.VCCPT INDEX=1, X
MAXPKTL=128, X
VWINDOW=1, * PACKET LEVEL WINDOW X
INSLOW=(100,50) * TO PREVENT SLOWDOWN

*
X25.VCCPT INDEX=2, X
MAXPKTL=128, X
VWINDOW=2, X
INSLOW=(100,50) * TO PREVENT SLOWDOWN

*
X25.VCCPT INDEX=3, X
MAXPKTL=128, X
VWINDOW=3, X
INSLOW=(100,50) * TO PREVENT SLOWDOWN

*
X25.VCCPT INDEX=4, X
MAXPKTL=1024, X
VWINDOW=127, X

```

## Example C (NPSI Only)

```

 INSLOW=(100,50) * TO PREVENT SLOWDOWN
*
X25.OUFT INDEX=1
*
X25.OUFT INDEX=2, * PKT SIZE=128,VWINDOW=2 X
 OPTFACL=420707430202
*
X25.OUFT INDEX=3, * PKT SIZE=128,VWINDOW=3, AND X
 OPTFACL=03430002420707430303, * REVERSECHARGING REQUESTX
 USRFILD=1234567890
*
X25.OUFT INDEX=4, * PKT SIZE=1024,VWINDOW=15 X
 OPTFACL=420A0A437F7F

*

* - MCH01 SL032: GATE FTPI HAVING TIMER TICK
* INCLUDED AT LINK EDIT
* 4 PVC 1 CTCP.
* 4 SVC 1 CTCP. (Selection via CUD0)

*
SL032 X25.MCH ADDRESS=032, X
 ACCOUNT=YES, X
 INTFAC=YES, X
 FRMLGTH=1028, X
 PKTMODL=128, X
 MWINDOW=7, LAP WINDOW X
 NPACOLL=(MCHLINE,MCHPU,VCPU), V3R4
 NPPVCN=8, V3R4
 NPADTEAD=42032, V3R4
 STATION=DTE, X
 LCGDEF=((0,3),(1,3)), X
 LCN0=USED, X
 FTPI=CUD0, X
 PUNAME=SP032, X
 VCID=YES, X
 GATE=GENERAL, X
 SUBADDR=NO, X
 SPEED=56000, X
 CUD0=(00,01), X
 CTCP=(00,01), X
 TAXUNIT=64, X
 LLCLIST=LLC4
*
 X25.LCG LCGN=0
*
*** 4 PVC FOR FTPI
*
 X25.VC LCN=(0,3), X25VC
 CTCN0=0, *
 HEXNAME=NO, *
 ISTATUS=ACTIVE, *
 LLC=LLC4, *
 NCPGRP=FTPIPVCs, *
 PRFLINE=FL032, *
 PRFPU=FP032, *
 SUFFIX=000, *

```

```

 TYPE=P, *
 VCCINDX=4
 EJECT
*
 X25.LCG LCGN=1
*
*** 4 SVC FOR FTPI
*
 X25.VC LCN=(0,3), X25VC
 HEXNAME=NO, *
 ISTATUS=ACTIVE, *
 CALL=INOUT, *
 NCPGRP=FTPISVCS, *
 PRFLINE=FL032, *
 PRFPU=FP032, *
 SUFFIX=128, *
 TYPE=S, *
 VCCINDX=4
* OUFINDX=4
 EJECT

 X25.END

CAGROUPD GROUP LNCTL=CA, X
 CA=TYPE6, X
 NCPCA=ACTIVE, X
 DELAY=0.0
CALINE1D LINE ADDRESS=00 * CHANNEL POSITION 0 LGETVXA3
CA01 PU PUTYPE=5 * USED for CUA=0C0 or 0D0
CALINE2D LINE ADDRESS=01 * CHANNEL POSITION 1 LGETVXA4
CA02 PU PUTYPE=5 * USED for CUA=0C0 or 0D0

 GENEND
 END

```

## Example D—NPSI HPTSS and Delayed Logical Link Activation

This section contains two examples that show the NCP and NPSI generation input, and the use of HPTSS and delayed logical link activation.

### NCP AND NPSI Generation input including JCL—Example D1

```

/*ROUTE PRINT LGEVMD.LCHARTRE
/*ROUTE PUNCH LGEVMD.LCHARTRE
//S1 EXEC PGM=IEHPRGM
//SYSPRINT DD SYSOUT=A
//DD1 DD VOL=SER=L28TS4,UNIT=3380,DISP=SHR
//SYSIN DD *
 SCRATCH DSNAME=NPSI.NCP103.NDFTBL,VOL=3380=L28TS5,MEMBER=ICNTABL1
 SCRATCH DSNAME=NPSI.NCP103.NDFTBL,VOL=3380=L28TS5,MEMBER=ICNTABL2
/*
//S2 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//DISK DD DISP=SHR,DSN=NPSI.NCP103.NDFTBL
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(10,1),,CONTIG)
//SYSUT4 DD UNIT=SYSDA,SPACE=(CYL,(10,1),,CONTIG)
//SYSIN DD *
 COPY OUTDD=DISK,INDD=DISK
/*
//STEP1 EXEC PGM=ICNRTNDF,REGION=6000K,PARM='LINECNT=45',TIME=1200
//STEPLIB DD DSN=LSTNCP.NB0903.SSPLIB,DISP=SHR 11.0
// DD DSN=NPSI.NDFLOAD.V3R5MVS,DISP=SHR
//GENDECK DD *

*** STAGE 1 INPUT
**
 OPTIONS USERGEN=(X25NPSI), X
 NEWDEFN=(YES,ECHO,SUPP)
*
BAKLOCAL PCCU DUMPDS=DUMPX25,SUBAREA=1
*
BAKBUILD BUILD NPA=(YES,DR), X
 VERSION=V6R1, * NCP VERSION X
 LOADLIB=LNCP11, X
 SESSACC=(YES,ALL,,750000,5000,,), X
 TYPGEN=NCP, X
 MODEL=3745-410, X
 NETID=FRIBM412, * HOST ID X
 MAXSUBA=63, X
 USGTIER=4, * 3,4,5-TD,TS,TB * 2-TS X
 X25.USGTIER=4, * X25.USGTIER X
 SUBAREA=44, X
 SLODOWN=50, X
 TRANSFR=100, X
 NUMHSAS=15, X
 BFRS=142, * BEST FOR NPSI PERFORMANCE 124 X
 LTRACE=8, * ALLOWS LINE TRACE ON FOUR MCH,S X
 OLT=YES, * note: 8 = max value for 3725 X
 DR3270=NO, X
 BRANCH=8000, X
 TRACE=(YES,100), * 100 ENTRIES IN ADDRESS TRACE TBL X
 NEWNAME=ST5244G, XX
 X25.PREFIX=A, * ALL NAMES START WITH S X

```

## Example D—HPTSS and Delayed Logical Link Activation

```

X25.IDNUMH=0F, * X25.IDNUMH MUST MATCH WITH SWITCHX
TYPYSYS=OS, * MAJOR NODE IDS.(2FIRST DIGITS) X
X25.SNAP=(4000,INACT), *PTIONAL DIAGNOSIS SNAP INCLUDED X
X25.MCHCNT=1, * NUMBER OF PHYSICAL LINKS X
X25.MAXPIU=5722

*
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD, X
 RIMM,NAKLIM,SESSION,SSPAUSE,XMTLMT)
SMMFTAB1 SWMMT NETID=FRIBM412,IDNUM=AAAAA,MAXOUT=7
BAKHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=1, X
 DELAY=0.1,TIMEOUT=30
 LUDRPOOL NUMTYP1=10, * FOR NON SNA CALLS X
 NUMTYP2=10 * FOR QLLC OR PSH CALLS

PATH41 PATH DESTSA=(1),ER0=(1,1),ER1=(1,1),ER2=(1,1) TO HOST
PATH42 PATH DESTSA=(5,2,3),ER0=(5,2),ER1=(5,3),ER2=(3,4)
 EJECT * AT LEAST ONE SDLC LINE
SDLC6 GROUP LNCTL=SDLC, * MUST BE DEFINED X
 TYPE=NCP, X
 DIAL=YES, X
 REPLYTO=20

SL6 LINE ADDRESS=9, * SDLC LINE FOR NIA FRONTAL X
 CLOCKNG=EXT, * X
 NEWSYNC=NO, X
 DUPLEX=HALF, X
 NPACOLL=YES, X
 NRZI=NO, X
 CALL=INOUT, X
 RETRIES=(7,0,0)
SP6 PU ISTATUS=INACTIVE, X
 PUTYPE=2

 EJECT
SDLC5 GROUP LNCTL=SDLC, X
 TYPE=NCP, X
 CLOCKNG=EXT, * X
 DIAL=NO, X
 REPLYTO=3, X
 LSPRI=NO, X
 DISCNT=NO, X
 ISTATUS=INACTIVE

SL5 LINE ADDRESS=10, * LEASED SDLC MULTIPOINT LINE X
 NEWSYNC=NO, X
 DUPLEX=HALF, X
 NPACOLL=YES, X
 NRZI=NO, X
 RETRIES=(3)
 SERVICE ORDER=SP5
SP5 PU ADDR=C2, * ADDRESS 3274 X
 MAXDATA=261, X
 MAXOUT=3, X
 PUTYPE=2,NPACOLL=NO, X
 VPACING=(63,63), X
 PACING=255
SU5 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,NPACOLL=NO

```

## Example D—HPTSS and Delayed Logical Link Activation

```

EJECT

GRNPA GROUP LNCTL=SDLC, * FOR NPM RECORDING X
 TYPE=NCP, X
 DIAL=NO, X
 DISCNT=NO, X
 VIRTUAL=YES, X
 ISTATUS=INACTIVE, X
 NPARSC=YES
LNNPA LINE LINEFVT=NPAVFVT
PUNPA PU PUTYPE=2, PUFVT=NPAPFVT
LUNPA LU LUFVT=(NPALFVT,NPALFVT),MAXCOLL=7, X
 VPACING=0,LOCADDR=1

EJECT

* NET1

*
NET1 X25.NET CPHINDX=3, X
 OUHINDX=3, X
 RFAC=OSIPRI, X
 NSTDFAC=(A1,A2,A3,A4), X
 INACTO=0295, X
 NETTYPE=4, X
 R20=00000020, X
 R22=000101, X
 R23=000000003, X
 T20=10, X
 T21=0050, X
 T22=00255, X
 T23=0201, X
 RESETINO=NO, X
 T25=3
VCCPT1 X25.VCCPT INDEX=1, X
 MAXPKTL=128, X
 INSLOW=(50,25), X
 VWINDOW=1
*
OUFT1 X25.OUFT INDEX=1
*

* MCH 1 *

*
MCH1 X25.MCH ADDRESS=1024, X
 ANS=CONT, X
 OWNER=(MARTINE,OLIVIA,SYLVIE), X
 PHYSRSC=YES, X
 XMONLNK=YES, X
 LLCLIST=(LLC0,LLC2,LLC3,LLC5), X
 ACCOUNT=NO, X
 TRANSFR=40, X
 NPADTEAD=, X
 LCGDEF=(0,15), X
 FRMLGTH=131, X

```

## Example D—HPTSS and Delayed Logical Link Activation

```

PKTMODL=8, X
VCCINDX=1, X
PAD=INTEG, X
TRAN=EVEN, X
PADCHECK=(YES), X
PLPIGGYB=NO, X
MBNOFULL=NO, X
SVCINN=1, X
SDRTCNT=1, X
SDRTIME=10, X
ACTIVTO=0.0, X
BRKCON=CONNECTO, X
MWINDOW=3, X
STATION=DTE, X
TPTIMER=25, X
TDTIMER=3, X
NPRETRY=3, X
NDRETRY=3, X
LUNAME=(CTXLU0), X
LCN0=USED, X
GATE=GENERAL, X
INTFAC=NO, X
SPAN=X2501, X
DBIT=NO, X
DEFPRI=NORMAL, X
SPEED=145000, X
DIRECT=NO

```

\*

```

X25.LCG LCGN=0
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC0,LCN=0,TRANSFR=4,X
 OWNER=(BUNNIE,CLIO)
X25.PU ISTATUS=INACTIVE,PUTYPE=2,ADDR=01,ANS=CONT
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC2,LCN=1
X25.PU PUTYPE=2,ISTATUS=INACTIVE,ADDR=01
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC3,LCN=3, X
 DSTNODE=INN,MONLINK=NO,TRANSFR=010
X25.PU ISTATUS=INACTIVE,PUTYPE=4
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC3,LCN=4, X
 DSTNODE=BNN,TRANSFR=010,OWNER=FANNIE
X25.PU ISTATUS=INACTIVE,ADDR=01
X25.LU LOCADDR=0
X25.LINE VCCINDX=1,TYPE=P,LLC=LLC4,LCN=5 OSITYPE=01
X25.PU ISTATUS=INACTIVE,ADDR=01
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC5,LCN=6
X25.PU ISTATUS=INACTIVE,ADDR=01
X25.VC VCCINDX=1,TYPE=P,SPAN=X2501,LCN=7,LLC=LLC0,ANS=CONT
X25.VC VCCINDX=1,TYPE=P,SPAN=X2501,LCN=8,LLC=LLC4,OSITYPE=00
X25.VC VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC5,LCN=9
X25.LINE VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=10
X25.PU PUTYPE=2,ISTATUS=INACTIVE,ANS=CONT
X25.VC VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=11,PRFLU=JOE
X25.VC VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=(12,13), X
 ANS=CONT
X25.LINE TYPE=S,SPAN=X2501,NCPGRP=CTXBIS,OWNER=(BARBARA), X
 DSTNODE=INN,MONLINK=YES,SWMMTID=SMMFTAB1
X25.PU ISTATUS=INACTIVE,PUTYPE=4
X25.VC VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=14, X
 NCPGRP=CTX1,OWNER=CAROLE

```

## Example D—HPTSS and Delayed Logical Link Activation

```

X25.LINE VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=15
X25.PU PUTYPE=2,ISTATUS=INACTIVE,ANS=CONT
*

*
* X25.END * ALB04100

* CHANNEL ADAPTER DEFINITION

*
CAGROUP GROUP LNCTL=CA, X
 CA=TYPE6, X
 DELAY=0.6, X
 NCPCA=ACTIVE
CALINE2 LINE ADDRESS=6
CA2PU PU PUTYPE=5
*
 GENEND ALB04200
 END ALB04300

//DBWORKFL DD UNIT=SYSDA,SPACE=(TRK,10)
//SYSPRINT DD SYSOUT=A
//PRINTER DD SYSOUT=A
//TBL1SRCE DD DSN=&&SRCE1,DISP=(,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,10)),DCB=BLKSIZE=3200
//TBL1LIST DD SYSOUT=P
//TBL1OBJ DD DSN=NPSI.NCP103.NDFTBL(ICNTABL1),DISP=OLD,
// VOL=SER=L28TS5,UNIT=3380
//TBL2SRCE DD DSN=&&SRCE2,DISP=(,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,10)),DCB=BLKSIZE=3200
//TBL2LIST DD SYSOUT=P
//TBL2OBJ DD DSN=NPSI.NCP103.NDFTBL(ICNTABL2),DISP=OLD,
// VOL=SER=L28TS5,UNIT=3380
//NEWDEFN DD SYSOUT=L
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10)),DISP=(,DELETE)
//SYSLIB DD DSN=NPSI.P61.MAC,DISP=SHR
// DD DSN=LSTNCP.NB0903.SNCPMAC1,DISP=SHR
//LNKSTMT DD DSN=NPSI.NCP103.NDFTBL(LINKED),DISP=SHR
/*

```



NCP Input Produced by NDF (NPSI)—Example D2

```

OPTIONS USERGEN=(X25NPSI),
NEWDEFN=(YES,ECHO,SUPP)
*
BAKLOCAL PCCU DUMPDS=DUMPX25,SUBAREA=1
*
BAKBUILD BUILD NPA=(YES,DR),
VERSION=V6R1, * NCP VERSION
LOADLIB=LNCP11,
SESSACC=(YES,ALL,,750000,5000,,),
TYPGEN=NCP,
MODEL=3745-410,
NETID=FRIBM412, * HOST ID
MAXSUBA=63,
USGTIER=4, * 3,4,5-TD,TS,TB * 2-TS
X25.USGTIER=4, * X25.USGTIER
SUBAREA=44,
SLODOWN=50,
TRANSFR=100,
NUMHSAS=15,
BFRS=142, * BEST FOR NPSI PERFORMANCE 124
LTRACE=8, * ALLOWS LINE TRACE ON FOUR MCH,S
OLT=YES, * note: 8 = max value for 3725
DR3270=NO,
BRANCH=8000,
TRACE=(YES,100), * 100 ENTRIES IN ADDRESS TRACE TBL
NEWNAME=ST5244G,
X25.PREFIX=A, * ALL NAMES START WITH S
X25.IDNUMH=0F, * X25.IDNUMH MUST MATCH WITH SWITCH*
TYP SYS=OS, * MAJOR NODE IDS.(2FIRST DIGITS)
X25.SNAP=(4000,INACT), *PTIONAL DIAGNOSIS SNAP INCLUDED
X25.MCHCNT=1, * NUMBER OF PHYSICAL LINKS
X25.MAXPIU=5722
*
BAKSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCOND,RECMD,RIMM,NAKLIM*,
,SESSION,SSPAUSE,XMTLMT)
SMMFTAB1 SWMMT NETID=FRIBM412,IDNUM=AAAAA,MAXOUT=7
BAKHOST HOST INBFRS=7,MAXBFRU=10,UNITSZ=196,BFRPAD=0,SUBAREA=1,
DELAY=0.1,TIMEOUT=30
LUDRPOOL NUMTYP1=10, * FOR NON SNA CALLS
NUMTYP2=10 * FOR QLLC OR PSH CALLS

PATH41 PATH DESTSA=(1),ER0=(1,1),ER1=(1,1),ER2=(1,1) TO HOST
PATH42 PATH DESTSA=(5,2,3),ER0=(5,2),ER1=(5,3),ER2=(3,4)
SDLC6 GROUP LNCTL=SDLC, * MUST BE DEFINED
TYPE=NCP,
DIAL=YES,
REPLYTO=20

SL6 LINE ADDRESS=9, * SDLC LINE FOR NIA FRONTAL
CLOCKNG=EXT, *
NEWSYNC=NO,
DUPLEX=HALF,
NPACOLL=YES,
NRZI=NO,
CALL=INOUT,

```

## Example D—HPTSS and Delayed Logical Link Activation

```

 RETRIES=(7,0,0)
SP6 PU ISTATUS=INACTIVE,
 PUTYPE=2

SDLC5 GROUP LNCTL=SDLC,
 TYPE=NCP,
 CLOCKNG=EXT,
 DIAL=NO,
 REPLYTO=3,
 LSPRI=NO,
 DISCNT=NO,
 ISTATUS=INACTIVE

SL5 LINE ADDRESS=10,
 NEWSYNC=NO,
 DUPLEX=HALF,
 NPACOLL=YES,
 NRZI=NO,
 RETRIES=(3)
 SERVICE ORDER=SP5
SP5 PU ADDR=C2,
 MAXDATA=261,
 MAXOUT=3,
 PUTYPE=2,NPACOLL=NO,
 VPACING=(63,63),
 PACING=255
SU5 LU LOCADDR=2,MODETAB=MODTD3,USSTAB=USSTD3,NPACOLL=NO

GRNPA GROUP LNCTL=SDLC,
 TYPE=NCP,
 DIAL=NO,
 DISCNT=NO,
 VIRTUAL=YES,
 ISTATUS=INACTIVE,
 NPARSC=YES
LNNPA LINE LINEFVT=NPAVFVT
PUNPA PU PUTYPE=2,PUFVT=NPAPFVT
LUNPA LU LUFVT=(NPALFVT,NPALFVT),MAXCOLL=7,
 VPACING=0,LOCADDR=1

* NET1

*
NET1 X25.NET CPHINDX=3,
 OUHINDX=3,
 RFAC=OSIPRI,
 NSTDFAC=(A1,A2,A3,A4),
 INACTO=0295,
 NETTYPE=4,
 R20=00000020,
 R22=000101,
 R23=000000003,
 T20=10,
 T21=0050,
 T22=00255,
 T23=0201,
 RESETINO=NO,

```

X2500390

## Example D—HPTSS and Delayed Logical Link Activation

```

 T25=3,
 DONE=YES
VCCPT1 X25.VCCPT INDEX=1,
 MAXPKTL=128,
 INSLW=(50,25),
 VWINDOW=1
*
OUFT1 X25.OUFT INDEX=1
*

*
* MCH 1
*

*
MCH1 X25.MCH ADDRESS=1024,
 ANS=CONT,
 OWNER=(MARTINE,OLIVIA,SYLVIE),
 PHYSRSC=YES,
 XMONLNK=YES,
 LLCLIST=(LLC0,LLC2,LLC3,LLC5),
 ACCOUNT=NO,
 TRANSFR=40,
 NPADTEAD=,
 LCGDEF=(0,15),
 FRMLGTH=131,
 PKTMODL=8,
 VCCINDX=1,
 PAD=INTEG,
 TRAN=EVEN,
 PADCHECK=(YES),
 PLPIGGYB=NO,
 MBNOFULL=NO,
 SVCINN=1,
 SDRTCNT=1,
 SDRTIME=10,
 ACTIVTO=0.0,
 BRKCON=CONNECTO,
 MWINDOW=3,
 STATION=DTE,
 TPTIMER=25,
 TDTIMER=3,
 NPRETRY=3,
 NDRETRY=3,
 LUNAME=(CTXLU0),
 LCN0=USED,
 GATE=GENERAL,
 INTFAC=NO,
 SPAN=X2501,
 DBIT=NO,
 DEFPRI=NORMAL,
 SPEED=145000,
 DIRECT=NO
*
X25.LCG LCGN=0
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC0,LCN=0,TRANSFR=4,*
 OWNER=(BUNNIE,CLIO)
X25.PU ISTATUS=INACTIVE,PUTYPE=2,ADDR=01,ANS=CONT

```

## Example D—HPTSS and Delayed Logical Link Activation

```
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC2,LCN=1
X25.PU PUTYPE=2,ISTATUS=INACTIVE,ADDR=01
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC3,LCN=3, *
 DSTNODE=INN,MONLINK=NO,TRANSFR=010
X25.PU ISTATUS=INACTIVE,PUTYPE=4
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC3,LCN=4, *
 DSTNODE=BNN,TRANSFR=010,OWNER=FANNIE
X25.PU ISTATUS=INACTIVE,ADDR=01
X25.LU LOCADDR=0
X25.LINE VCCINDX=1,TYPE=P,LLC=LLC4,LCN=5 OSITYPE=01
X25.PU ISTATUS=INACTIVE,ADDR=01
X25.LINE VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC5,LCN=6
X25.PU ISTATUS=INACTIVE,ADDR=01 NPACOLL=NO
X25.VC VCCINDX=1,TYPE=P,SPAN=X2501,LCN=7,LLC=LLC0,ANS=CONT
X25.VC VCCINDX=1,TYPE=P,SPAN=X2501,LCN=8,LLC=LLC4,OSITYPE=00
X25.VC VCCINDX=1,TYPE=P,SPAN=X2501,LLC=LLC5,LCN=9
X25.LINE VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=10
X25.PU PUTYPE=2,ISTATUS=INACTIVE,ANS=CONT
X25.VC VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=11,PRFLU=JOE
X25.VC VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=(12,13), *
 ANS=CONT
X25.LINE TYPE=S,SPAN=X2501,NCPGRP=CTXBIS,OWNER=(BARBARA), *
 DSTNODE=INN,MONLINK=YES,SMMTID=SMMFTAB1
X25.PU ISTATUS=INACTIVE,PUTYPE=4
X25.VC VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=14, *
 NCPGRP=CTX1,OWNER=CAROLE
X25.LINE VCCINDX=1,TYPE=S,SPAN=X2501,OUFINDX=1,LCN=15
X25.PU PUTYPE=2,ISTATUS=INACTIVE,ANS=CONT
```

## Example D—HPTSS and Delayed Logical Link Activation

```
X25.END
IGNORE * LINES BETWEEN IGNORE/NOIGNORE WILL BE REGENERATED

* THE REUSE OPTION ON NEWDEFN KEYWORD WAS PROCESSED IN THIS GENERATION*
* DEFINITION. GENERATION STATEMENTS LOCATED IN BETWEEN AND INCLUDING *
* THE IGNORE-NOIGNORE PAIRS MUST NOT BE CHANGED IN ANY WAY. *

A25PA24A GROUP DIAL=NO, *
 PHYRSC=APA24, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALPBDT,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
ALA24000 LINE SPAN=X2501, *
 OWNER=(BUNNIE,CLIO), *
 UACB=XAA24000
ACA24000 SERVICE
APA24000 PU ADDR=01, *
 ANS=CONT, *
 ISTATUS=INACTIVE, *
 PUTYPE=2, *
 XID=NO, *
 PUDR=NO
ALA24001 LINE SPAN=X2501, *
 UACB=XAA24001
ACA24001 SERVICE
APA24001 PU ADDR=01, *
 ISTATUS=INACTIVE, *
 PUTYPE=2, *
 XID=NO, *
 PUDR=NO
A25PA24B GROUP DIAL=NO, *
 PHYRSC=APA24, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALPBDT,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
ALA24003 LINE SPAN=X2501, *
```

## Example D—HPTSS and Delayed Logical Link Activation

```

 TRANSFR=010,
 MONLINK=NO,
 UACB=XAA24003
ACA24003 SERVICE
APA24003 PU ISTATUS=INACTIVE,
 PUTYPE=4,
 PUDR=NO
A25PA24C GROUP DIAL=NO,
 PHYRSC=APA24,
 LEVEL2=BALNAVL2,
 LEVEL3=BALNAVL3,
 LEVEL5=NCP,
 LINEADD=NONE,
 COMPTAD=NO,
 COMPOWN=YES,
 COMPSWP=NO,
 COMPACB=NO,
 LNCTL=SDLC,
 NPACOLL=NO,
 TIMER=(BALNATER,,BALNATST,BALNATLS),
 TYPE=NCP,
 USERID=(5688035,BALPBDT,NORECMS),
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
ALA24004 LINE SPAN=X2501,
 TRANSFR=010,
 OWNER=FANNIE,
 UACB=XAA24004
ACA24004 SERVICE
APA24004 PU ADDR=01,
 ISTATUS=INACTIVE,
 PUTYPE=1,
 PUDR=NO
AUA24004 LU LOCADDR=0,
 LUDR=NO
ALA24005 LINE UACB=XAA24005
ACA24005 SERVICE
APA24005 PU ADDR=01,
 ISTATUS=INACTIVE,
 PUTYPE=1,
 XID=NO,
 MAXOUT=6,
 PUDR=NO
ALA24006 LINE SPAN=X2501,
 UACB=XAA24006
ACA24006 SERVICE
APA24006 PU ADDR=01,
 ISTATUS=INACTIVE,
 PUTYPE=1,
 XID=NO,
 PUDR=NO
ALA24007 LINE IPL=NO,
 SPAN=X2501,
 UACB=XAA24007
ACA24007 SERVICE
APA24007 PU ADDR=01,
 PUDR=NO,
 PUTYPE=1,
 XID=NO,
```

## Example D—HPTSS and Delayed Logical Link Activation

```

 SPAN=X2501, *
 ANS=CONT, *
 MAXDATA=265
AUA24007 LU LOCADDR=0, *
 SPAN=X2501, *
 VPACING=(2,1), *
 PACING=1, *
 LUDR=NO
ALA24008 LINE IPL=NO, *
 SPAN=X2501, *
 UACB=XAA24008
ACA24008 SERVICE
APA24008 PU ADDR=01, *
 PUDR=NO, *
 PUTYPE=1, *
 XID=NO, *
 MAXOUT=6, *
 SPAN=X2501, *
 MAXDATA=265
AUA24008 LU LOCADDR=0, *
 SPAN=X2501, *
 VPACING=(2,1), *
 PACING=1, *
 LUDR=NO
ALA24009 LINE IPL=NO, *
 SPAN=X2501, *
 UACB=XAA24009
ACA24009 SERVICE
APA24009 PU ADDR=01, *
 PUDR=NO, *
 PUTYPE=1, *
 XID=NO, *
 SPAN=X2501, *
 MAXDATA=265
AUA24009 LU LOCADDR=0, *
 SPAN=X2501, *
 VPACING=(2,1), *
 PACING=1, *
 LUDR=NO
CTX1 GROUP DIAL=YES, *
 PHYRSC=APA24, *
 OWNER=CAROLE, *
 LEVEL2=BALNAVL2, *
 LEVEL3=BALNAVL3, *
 LEVEL5=NCP, *
 LINEADD=NONE, *
 COMPTAD=NO, *
 COMPOWN=YES, *
 COMPSWP=NO, *
 COMPACB=NO, *
 LINEAUT=YES, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALNATER,,BALNATST,BALNATLS), *
 TYPE=NCP, *
 USERID=(5688035,BALSBTD,NORECMS), *
 XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXX)
ALA2400F LINE SPAN=X2501, *

```

## Example D—HPTSS and Delayed Logical Link Activation

```

CALL=INOUT, *
TRANSFR=40, *
UACB=XAA2400F
APA2400F PU ISTATUS=INACTIVE, *
ANS=CONT, *
PUTYPE=2
ALA2400E LINE CALL=INOUT, *
TRANSFR=40, *
SPAN=X2501, *
UACB=XAA2400E
APA2400E PU SPAN=X2501, *
PUTYPE=(1,2)
CTXBIS GROUP DIAL=YES, *
PHYSRSC=APA24, *
OWNER=(BARBARA), *
LEVEL2=BALNAVL2, *
LEVEL3=BALNAVL3, *
LEVEL5=NCP, *
LINEADD=NONE, *
COMPTAD=NO, *
COMPOWN=YES, *
COMPSWP=NO, *
COMPACB=NO, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALNATER,,BALNATST,BALNATLS), *
TYPE=NCP, *
USERID=(5688035,BALIBDT,NORECMS), *
PUTYPE=4, *
SHM=YES, *
ACTIVTO=0.0, *
BRKCON=CONNECTO, *
XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
ALA24GGH LINE SPAN=X2501, *
SMMTID=SMMFTAB1, *
CALL=INOUT, *
TRANSFR=40, *
MONLINK=YES, *
UACB=XAA24GGH
APA24GGH PU ISTATUS=INACTIVE, *
PUTYPE=4
A25SA24D GROUP DIAL=YES, *
PHYSRSC=APA24, *
LEVEL2=BALNAVL2, *
LEVEL3=BALNAVL3, *
LEVEL5=NCP, *
LINEADD=NONE, *
COMPTAD=NO, *
COMPOWN=YES, *
COMPSWP=NO, *
COMPACB=NO, *
LINEAUT=YES, *
LNCTL=SDLC, *
NPACOLL=NO, *
TIMER=(BALNATER,,BALNATST,BALNATLS), *
TYPE=NCP, *
USERID=(5688035,BALSBDT,NORECMS), *

```



## Example D—HPTSS and Delayed Logical Link Activation

```

XIO=(BALNAVXL,BALNAVXS,BALNAVXI,BALNAVXK)
ALA2400D LINE CALL=INOUT, *
 TRANSFR=40, *
 SPAN=X2501, *
 UACB=XAA2400D
APA2400D PU SPAN=X2501, *
 ANS=CONT, *
 PUTYPE=(1,2)
ALA2400C LINE CALL=INOUT, *
 TRANSFR=40, *
 SPAN=X2501, *
 UACB=XAA2400C
APA2400C PU SPAN=X2501, *
 ANS=CONT, *
 PUTYPE=(1,2)
ALA2400B LINE CALL=INOUT, *
 TRANSFR=40, *
 SPAN=X2501, *
 UACB=XAA2400B
APA2400B PU SPAN=X2501, *
 PUTYPE=(1,2)
ANET11 GROUP DIAL=NO, *
 BERPROC=BALNUMBER, *
 COMPACB=YES, *
 COMPTAD=YES, *
 COMPOWN=YES, *
 COMPSWP=YES, *
 LEVEL2=BALNAML2, *
 LEVEL3=BALNAML3, *
 LEVEL5=NCP, *
 LNCTL=SDLC, *
 NPACOLL=NO, *
 TIMER=(BALLAP4,,BALLAP4,BALLAP4), *
 TYPE=NCP, *
 USERID=(5688035,BALMBDT,NORECMS,P), *
 XIO=(BALNAMXL,BALNAMXS,BALNAMXI,BALNAMXK)
MCH1 LINE OWNER=(MARTINE,OLIVIA,SYLVIE), *
 ADDRESS=(1024,FULL), *
 SPAN=X2501, *
 XMONLNK=YES, *
 UACB=(X25AA24X,X25AA24R)
ACA24 SERVICE
APA24 PU ADDR=01, *
 ANS=CONT, *
 SPAN=X2501, *
 MAXDATA=265, *
 PUDR=NO, *
 XID=NO, *
 PUTYPE=1
CTXLU0 LU LOCADDR=0,SPAN=X2501,VPACING=(2,1),ISTATUS=ACTIVE
NOIGNORE * LINES BETWEEN IGNORE/NOIGNORE WILL BE REGENERATED

* CHANNEL ADAPTER DEFINITION

*
CAGROUP GROUP LNCTL=CA, *
 CA=TYPE6, *
 DELAY=0.6, *
```

## Example D—HPTSS and Delayed Logical Link Activation

```

 NCPA=ACTIVE
CALINE2 LINE ADDRESS=6
CA2PU PU PUTYPE=5
*
 GENEND INIT=BALINIMD,
 UGLOBAL=BALNMGOP

```

**Example E—X.25/ISDN Interworking (NPSI V3R6 and Later)**

This section contains an example that shows the NCP and NPSI generation input, which contains the ISDN interworking function.

```

*
* A 7 1 I S D N *
*

*
* NCP NAME: A71ISDN *
* NCP VER/REL: V6R2 *
* NPSI VER/REL: V3R6 *
* BOX-TYPE: 3745 *
*
*
* CONTENTS: THIS GEN CONTAINS THE FOLLOWING: *
*
* ADDR BRIEF DESCRIPTION *
* ---- -
* 001 MCH1 PCNE 9600 LINE *
*
*

* GENERATION OF THE X25 BLOCKS FOR NEO WITHIN NCP *
* X.25 NPSI MCH DESCRIPTION *

*
* OPTIONS NEWDEFN=(YES,ECHO),USERGEN=(X25NPSI,FNMNDFGN)
*
APCCU1 PCCU AUTOSYN=YES, X
 BACKUP=YES,
 CDUMPDS=CSPDUMP,
 CONFGDS=CRNCKPT,
 DUMPDS=VTAMDUMP,
 MAXDATA=4096,
 MDUMPDS=MOSSDUMP,
 NETID=NETA,
 OWNER=A01N,
 SUBAREA=01
*

* NCP BUILD MACRO - NCP/CONTROLLER INFO *

*
A71ISDN BUILD ADSESS=250,
 AUXADDR=250,
 BACKUP=500,
 BFRS=104,
 BRANCH=8000,
 CATRACE=(YES,255),
 CSMSG=C3D9C9E340E2C9E340D4C5E2E2C1C7C540C6D6D940E2E24040
 40C2C340E3C5D9D4C9D5C1D3,
 CWALL=26,
 ENABLTO=30.0,
 ERASE=YES,

```

## Example E—X.25/ISDN Interworking

```
ERLIMIT=16,
LOADLIB=NCPLOAD,
LTRACE=8,
MAXSESS=250,
MAXSSCP=8,
MODEL=3745,
NAMTAB=30,
NETID=NETA,
NEWNAME=A71ISDN,
NPA=(YES,DR),
NUMHSAS=8,
OLT=YES,
PWROFF=YES,
SALIMIT=511,
SESSACC=(YES,ALL,,256,64,,10,30,50,100,150,200),
SLODOWN=12,
SUBAREA=71,
TRACE=(YES,100),
TRANSFR=41,
TWXID=(E8D6E4C3C1D3D311,C2C9C7D5C3D7C3C1D3D325),
TYPGEN=NCP,
TYP SYS=MVS,
USGTIER=5,
X25.USGTIER=5, X.25 NPSI DEFINITIONS
X25.BYTTTHROD=100000, MAX BYTE THRESHOLD
X25.SEGTHROD=100000, MAX SEGMENT THRESHOLD
X25.SNAP=4000, 4K SNAP Trace Allowed
X25.IDNUMH=2, 1ST DIGIT OF SMN IDNUM
X25.MCHCNT=2, 2 MCH IN THIS GENERATION
X25.PAHINDX=6, LARGEST INDX VALUE FOR X25.PAD
X25.PREFIX=A, ADDRESS PREFIX = 'A'
X25.MAXPIU=64K,
VERSION=V6R2,
VRPOOL=100
```

\*

```

* DYNAMIC CONTROL FACILITIES USED BY VTAM *

```

\*

```
SYSCNTRL OPTIONS=(BACKUP,
BHSASSC,
DLRID,
DVSINIT,
ENDCALL,
LNSTAT,
MODE,
NAKLIM,
RCNTRL,
RCOND,
RDEVQ,
RECMD,
RIMM,
SESINIT,
SESSION,
SSPAUSE,
STORDSP,
```

```

XMTLMT)
*

* SMMF Tables

*
SMMFTAB1 SWMMT IDNUM=AAAA1,
 NETID=NETA,
 MAXOUT=7
*
SMMFTAB2 SWMMT IDNUM=AAAA2,
 NETID=NETA,
 MAXOUT=7
*

* NCP HOST MACRO - CHANNEL ATTACHED HOST DEFINITIONS

*
A01N HOST BFRPAD=0,
 INBFRS=6,
 MAXBFRU=16,
 NETID=NETA,
 SUBAREA=01,
 UNITSZ=256
*

* PATHS

*
 PATH DESTSA=1,
 ER0=(1,1,5000,5000,5000,20000),
 ER1=(1,1),
 ER2=(1,1),
 ER3=(1,1),
 VR0=0,
 VRPWS00=(9,200),VRPWS01=(9,200),VRPWS02=(9,200),
 VR1=1,
 VRPWS10=(9,200),VRPWS11=(9,200),VRPWS12=(9,200),
 VR2=2,
 VRPWS20=(9,200),VRPWS21=(9,200),VRPWS22=(9,200),
 VR3=3,
 VRPWS30=(9,200),VRPWS31=(9,200),VRPWS32=(9,200)
*
*

* PUDR and LUDR Pools

*
 PUDRPOOL NUMBER=5
*
 LUDRPOOL NUMTYP1=50,
 NUMTYP2=100,
 NUMILU=500
*

```

## Example E—X.25/ISDN Interworking

```

* SDLC LEASED 3276 LU TYPE 2 SDLC LINE 11 *

*
EGP3276 GROUP LNCTL=SDLC,
 TYPE=NCP,
 DIAL=NO,
 REPLYTO=3
*
EL13276 LINE ADDRESS=4,CLOCKNG=EXT,DUPLEX=FULL,SPEED=2400, NEW
 PUDR=YES,MAXPU=4,NRZI=NO,LPDATS=LPDA1
*
 SERVICE ORDER=(EP13276),MAXLIST=14
*
EP13276 PU ADDR=01, PU: NAME MUST MATCH TCAM
 MAXDATA=261, CONTROL INFOR PLUS 256 DATA
 MAXOUT=7, DR INCLUDED PU MAY BE DELETED
 PUDR=YES, PUTYPE=2,
 PASSLIM=9 ALLOW FULL SCREEN WRITE (9X256)
*
EU13276 LU LOCADDR=2,PACING=(1) 3277 LU: NAME MUST MATCH TCAM
EU23276 LU LOCADDR=3,PACING=(1) 3277 LU: NAME MUST MATCH TCAM
EU33276 LU LOCADDR=4,PACING=(1) 3277 LU: PRINTER "*****"
*

* NPA Definitions

*
NPAGD GROUP LNCTL=SDLC, * FOR NPM RECORDING
 TYPE=NCP,
 DIAL=NO,
 DISCNT=NO,
 VIRTUAL=YES,
 NPARSC=YES,
 ISTATUS=ACTIVE
*
NPADLN LINE
NPADPU PU MODETAB=X25MOTAB
NPADLU1 LU DLOGMOD=LU2V3P2
NPADLU2 LU DLOGMOD=LU2V6P2
NPADLU3 LU DLOGMOD=LU2V7P2
*

*
* FIRST NETWORK TRANSPAC
*

*
TRANSPAC X25.NET CPHINDX=4,
 DM=YES,
 CAUSE=CCITT,
 NETTYPE=1,
 NSTDFAC=(49),
 OUHINDX=5,
 RESETINO=(0085,0086,8183,8184,8185,8186,8188,8189,8987,X
 X82,8FXX,0FXX) * THESE RESET CC DD WILL NOT INOP
*

```

## Example E—X.25/ISDN Interworking

```

X25.VCCPT INDEX=1,MAXPKTL=128,VWINDOW=1
X25.VCCPT INDEX=2,MAXPKTL=128,VWINDOW=2
X25.VCCPT INDEX=3,MAXPKTL=128,VWINDOW=3
*
X25.OUFT INDEX=1
X25.OUFT INDEX=2,OPTFACL=420707430202,USRFILD=1234567890
X25.OUFT INDEX=3,OPTFACL=03690102420707430303,
 USRFILD=1234567890
X25.OUFT INDEX=4,OPTFACL=420A0A431515
*

* LENGTH=109

*
X25.OUFT INDEX=5,
 OPTFACL=034442050501020355430707430606010343050543060643
 04044204040366420505420505420505420505420505420505420505
 42050542050501024306064306064306064306064205050355035503
 5503554204044204040102010201024306060344034403444205
 05,
 USRFILD=1234567890
*

* MCH one *
* *
* EL128 = MCH INTERWORKING for dial-in-by-the-DTE operations *
* V35 Physical interface 64Kbps to be connected to 7820 *
* *
* NO GATE MCH LINE PU LU *
* NO DATE EL128 EP128 EU128 *
* GROUP EG128 *
*C062? POOL of 16 SVCs SWITCHED GROUP: EG128SVC *
* 0-01 SVC -OUT XL128001 *
* BLK2A1?? 0-02 SVC -OUT XL128002 XP128002 *
* 0-03 SVC -OUT XL128003 XP128003 XU128003 *
* 0-04 SVC -OUT XL128004 XP128004 XU128004 *

*
AL016 X25.MCH ADDRESS=016,
 ACCOUNT=(LLC0,LLC3B,LLC5),
 ANS=CONT,
 DBIT=YES,
 CCXDELAY=NONE,
 FRMLGTH=131,
 GATE=NO, * GATE NO SUPPORTED
 IDBLKC=062, * PCNE ID BLOCK
 IDBLKP=064, * PAD ID BLOCK
 LCGDEF=(0,10),
 LCN0=NOTUSED,
 LLCLIST=(LLC0,LLC3,LLC5),
 LSPRI=NO,
 LUNAME=AU16,
 MBITCHN=YES,
 MWINDOW=3,
 NCPGRP=AG16,
 NDRETRY=3,
 NPACOLL=(MCHLINE,MCHPU,VCPU),

```

## Example E—X.25/ISDN Interworking

```

NPPVCN=1,
NPADTEAD=4216,
NPRETRY=10,
PAD=INTEG, * INTEGRATED PAD
PKTMODL=8,
PUNAME=AP16,
ROLE=NEG,
TDTIMER=3,
TPTIMER=2,
XMONLNK=NO,
SPEED=64000, * MANDATORY LINE SPEED
STATION=DTE,
PHYSRSC=NO, * NOT SUPPORTED
INTWLINK=DIALDTE * CALLS FROM NPSI TO PSPDN
*
X25.LCG LCGN=0
*

* 16 X25 SWITCHED VCs OUTGOING for dial-in-by-the-DTE operations
* TRANSPAC connections to EBS64 (BNN only)

*
X25.VC CALL=OUT,
ISTATUS=INACTIVE,
IDNUMT=A102,
LCN=(1,10),
OUFINDX=1,
NCPGRP=AG016SVC,
SPAN=OPER1,
TYPE=S,
RETVCTO=1,
VCCINDX=2
*

* MCH two *
* *
* EL144 = MCH INTERWORKING FOR DIAL-OUT-BY-THE-PSPDN OPERATIONS *
* V35 PHYSICAL INTERFACE 64KBPS TO BE CONNECTED TO 7820 *
* *
* GATE LINK PU LU *
* EL144 EP144 EU144 *
* GROUP EG144 *
* *
* 1 SVC Switched GROUP: XG144SVC *
* 0-01 SVC IN XL144001 XP144001 XL144001 *

*
AL020 X25.MCH ADDRESS=020,
ACCOUNT=(LLC0,LLC3B,LLC5),
ANS=CONT,
DBIT=YES,
CCXDELAY=NONE,
FRMLGTH=131,
GATE=GENERAL, * GATE SUPPORTED
IDBLKC=062, * PCNE ID BLOCK
IDBLKP=064, * PAD ID BLOCK
LCGDEF=(0,1),

```



## Example E—X.25/ISDN Interworking

```

LCN0=NOTUSED,
LLCLIST=(LLC0,LLC3,LLC4,LLC5),
LSPRI=NO,
LUNAME=AU020,
MBITCHN=YES,
MWINDOW=3,
NCPGRP=AG020,
NDRETRY=3,
NPACOLL=(MCHLINE,MCHPU,VCPU),
NPPVCN=1,
NPADTEAD=42020,
NPRETRY=10,
PAD=INTEG,
PKTMODL=8,
PUNAME=AP020,
ROLE=NEG,
TDTIMER=3,
TPTIMER=2,
XMONLNK=NO,
SPEED=64000,
STATION=DCE,
INTWLINK=DIALPSDN
* MANDATORY SPEED
*
* CALL IN FROM PSPDN
*
*
X25.LCG LCGN=0
*

* 1 X25 SWITCHED VC INCOMING for dial-out-by-the-PSPDN operations
* TRANSPAC connections from SBS64 (BNN only)

*
X25.VC CALL=IN,
ISTATUS=INACTIVE,
IDNUMT=B102,
LCN=1,
OUFINDX=1,
NCPGRP=AG020SVC,
SPAN=OPER1,
TYPE=S,
RETVCTO=1,
VCCINDX=2
*
X25.END
*

* CHANNEL ADAPTER DEFINITIONS *

*
A71XCA0 GROUP LNCTL=CA,
CA=TYPE6,
NCPCA=ACTIVE,
TIMEOUT=180.0
*

* CA ADDR - 00 PHYSICAL PORT 5 *

*
A71C00 LINE ADDRESS=00,

```

## Example E—X.25/ISDN Interworking

```

 CASDL=420,
 INBFRS=3,
 DELAY=0.2
*
A71P00A PU PUTYPE=5,
 TGN=1
*

* NCP GENEND MACRO - END OF GEN *

*
GENEND GENEND
 END
```

## Example F—Generation Information for X.25 ODL

This section contains an example that shows NCP generation input for X.25 ODL resources.

```

* A 7 1 S O X 2 *

* * *
* NCP NAME: A71SOX2 - ODL X.25 * *
* VER/REL: V7R4F * *
* SUBAREA: 71 * *
* BOX-TYPE: 3745/900 * *
* * *

* NCP OPTIONS MACRO * *

* * *
* OPTIONS NEWDEFN=(YES,ECHO),USERGEN=(CBEX25,FNMNDFGN) *
* OPTIONS NEWDEFN=(YES,ECHO),USERGEN=(CBEX25,FNMNDFGN,ATFTUNE) *
* * *

* VTAM PCCU MACRO * *

* * *
APCCU1 PCCU AUTOSYN=YES,BACKUP=YES,CDUMPDS=CSPDUMP,CONFIGDS=CRNCKPT, X
 DUMPDS=VTAMDUMP,GWCTL=SHR,MAXDATA=8192,MDUMPDS=MOSSDUMP,X
 OWNER=A01N,SUBAREA=01
*
APCCU3 PCCU AUTOSYN=YES,BACKUP=YES,CDUMPDS=CSPDUMP,CONFIGDS=CRNCKPT, X
 DUMPDS=VTAMDUMP,GWCTL=SHR,MAXDATA=8192,MDUMPDS=MOSSDUMP,X
 OWNER=A03N,SUBAREA=03
*

* NCP BUILD MACRO - NCP/CONTROLLER INFO *

* * *
A71SOX2 BUILD ADSESS=5, X
 AUXADDR=5, X
 BACKUP=500, X
 BFRS=128, X
 BRANCH=8000, X
 CATRACE=(YES,255), X
 CSMG=C3D9C9E340E2C9E340D4C5E2E2C1C7C540C6D6D940E2E24040X
 40C2C340E3C5D9D4C9D5C1D3, X
 CWALL=26, X
 DYNPOOL=(79,78), X
 ERASE=YES, X
 ERLIMIT=16, X
 LOADLIB=NCPLoad, X
 LTRACE=8, X
 MAXSESS=250, X
 MAXSSCP=8, X
 MAXSUBA=100, X
 MODEL=3745, X
 MXRLINE=04, X
 MXVLINE=60, X
 NAMTAB=3, X

```

## Example F—Gen Information for X.25 ODLC

```

NETID=NETA, X
NEWNAME=A71SOX2, X
NPA=(YES,DR), X
SESSACC=(YES,ALL,,256,64,,10,30,50,100,150,200), X
NUMHSAS=100, X
OLT=YES, X
PWROFF=YES, X
SALIMIT=255, X
SLODOWN=12, X
SUBAREA=71, X
TRACE=(YES,100), X
TRANSFR=32, X
TWXID=(E8D6E4C3C1D3D311,C2C9C7D5C3D7C3C1D3D325), X
TYPGEN=NCP, X
TYP SYS=MVS, X
USGTIER=5, X
VERSION=V7R4F, NCP Version X
VRPOOL=1000, X
VRTIMER0=(10,10), X
VRTIMER1=(10,10), X
VRTIMER2=(10,10), X
X25.USGTIER=5, X.25 NPSI DEFINITIONS X
X25.BYTTTHROD=100000, MAX BYTE THRESHOLD X
X25.SEGTHROD=100000, MAX SEGMENT THRESHOLD X
X25.SNAP=4000, 4K Snap Trace for DCR X
X25.MCHCNT=1, 1 MCH CODED IN THIS GEN X
X25.PREFIX=X, ADDRESS PREFIX = 'X' X
X25.MAXPIU=64K, X
X25.PAHINDX=6, X
X25.IDNUMH=8 1ST DIGIT OF IDNUM IN SMN X
*

* DYNAMIC CONTROL FACILITIES USED BY VTAM *

*
SYSCNTRL OPTIONS=(BACKUP,BHSASSC,DLRID,DVSINIT,ENDCALL, X
LNSTAT,MODE,NAKLIM,RCNTRL,RCOND,RDEVQ,RECMD,RIMM, X
SESINIT,SESSION,SSPAUSE,STORDSP,XMTLMT) X
*

* NCP HOST MACRO - CHANNEL ATTACHED HOST DEFINITIONS *

*
A01N HOST BFRPAD=0,INBFRS=6,MAXBFRU=32,SUBAREA=01, X
UNITSZ=256 X
*
A03N HOST BFRPAD=0,INBFRS=6,MAXBFRU=32,SUBAREA=03, X
UNITSZ=256 X

* PATH TO OWNING HOST *

*
PATH DESTSA=01, X
ER0=(01,1),ER1=(01,1),ER2=(01,1),ER3=(01,1), X
ER4=(01,1),ER5=(01,1),ER6=(01,1),ER7=(01,1), X
ER8=(01,1),ER9=(01,1), X
VR0=0, X
VRPWS00=(9,200),VRPWS01=(9,200),VRPWS02=(9,200), X

```

## Example F—Gen Information for X.25 ODL

|      |                                                      |   |
|------|------------------------------------------------------|---|
|      | VR1=1,                                               | X |
|      | VRPWS10=(9,200),VRPWS11=(9,200),VRPWS12=(9,200),     | X |
|      | VR2=2,                                               | X |
|      | VRPWS20=(9,200),VRPWS21=(9,200),VRPWS22=(9,200),     | X |
|      | VR3=3,                                               | X |
|      | VRPWS30=(9,200),VRPWS31=(9,200),VRPWS32=(9,200),     | X |
|      | VR4=4,                                               | X |
|      | VRPWS40=(9,200),VRPWS41=(9,200),VRPWS42=(9,200),     | X |
|      | VR5=5,                                               | X |
|      | VRPWS50=(9,200),VRPWS51=(9,200),VRPWS52=(9,200),     | X |
|      | VR6=6,                                               | X |
|      | VRPWS60=(9,200),VRPWS61=(9,200),VRPWS62=(9,200),     | X |
|      | VR7=7,                                               | X |
|      | VRPWS70=(9,200),VRPWS71=(9,200),VRPWS72=(9,200)      | X |
| *    |                                                      |   |
| PATH | DESTSA=03,                                           | X |
|      | ER0=(03,1),ER1=(03,1),ER2=(03,1),ER3=(03,1),         | X |
|      | ER4=(03,1),ER5=(03,1),ER6=(03,1),ER7=(03,1),         | X |
|      | ER8=(03,1),ER9=(03,1),                               | X |
|      | VR0=0,                                               | X |
|      | VRPWS00=(9,200),VRPWS01=(9,200),VRPWS02=(9,200),     | X |
|      | VR1=1,                                               | X |
|      | VRPWS10=(9,200),VRPWS11=(9,200),VRPWS12=(9,200),     | X |
|      | VR2=2,                                               | X |
|      | VRPWS20=(9,200),VRPWS21=(9,200),VRPWS22=(9,200),     | X |
|      | VR3=3,                                               | X |
|      | VRPWS30=(9,200),VRPWS31=(9,200),VRPWS32=(9,200),     | X |
|      | VR4=4,                                               | X |
|      | VRPWS40=(9,200),VRPWS41=(9,200),VRPWS42=(9,200),     | X |
|      | VR5=5,                                               | X |
|      | VRPWS50=(9,200),VRPWS51=(9,200),VRPWS52=(9,200),     | X |
|      | VR6=6,                                               | X |
|      | VRPWS60=(9,200),VRPWS61=(9,200),VRPWS62=(9,200),     | X |
|      | VR7=7,                                               | X |
|      | VRPWS70=(9,200),VRPWS71=(9,200),VRPWS72=(9,200)      | X |
| *    |                                                      |   |
| PATH | DESTSA=(02,04),                                      | X |
|      | ER0=(04,1),ER1=(04,2),ER2=(04,3),ER3=(04,4),         | X |
|      | ER4=(04,5),ER5=(04,6),ER6=(04,7),ER7=(04,8),         | X |
|      | ER8=(04,9),ER9=(04,10),                              | X |
|      | ER10=(04,11),ER11=(04,12),ER12=(04,13),ER13=(04,14), | X |
|      | VR0=0,                                               | X |
|      | VRPWS00=(9,200),VRPWS01=(9,200),VRPWS02=(9,200),     | X |
|      | VR1=1,                                               | X |
|      | VRPWS10=(9,200),VRPWS11=(9,200),VRPWS12=(9,200),     | X |
|      | VR2=2,                                               | X |
|      | VRPWS20=(9,200),VRPWS21=(9,200),VRPWS22=(9,200),     | X |
|      | VR3=3,                                               | X |
|      | VRPWS30=(9,200),VRPWS31=(9,200),VRPWS32=(9,200),     | X |
|      | VR4=4,                                               | X |
|      | VRPWS40=(9,200),VRPWS41=(9,200),VRPWS42=(9,200),     | X |
|      | VR5=5,                                               | X |
|      | VRPWS50=(9,200),VRPWS51=(9,200),VRPWS52=(9,200),     | X |
|      | VR6=6,                                               | X |
|      | VRPWS60=(9,200),VRPWS61=(9,200),VRPWS62=(9,200),     | X |
|      | VR7=7,                                               | X |
|      | VRPWS70=(9,200),VRPWS71=(9,200),VRPWS72=(9,200)      | X |

\*

## Example F—Gen Information for X.25 ODLC

```

* NCP POOL MACROS - DYN RECONFIG & SWITCHED SDLC LINKS *

*
* PUDRPOOL NUMBER=1
* LUDRPOOL NUMILU=5,NUMTYP1=1,NUMTYP2=5
*

* NPA DEFINITION - NETWORK PERFORMANCE ANALYZER *

*
A71NPAG GROUP LNCTL=SDLC, X
 NPARSC=YES, X
 VIRTUAL=YES
*
A71NPAL LINE
*
A71NPPU PU
*
A71NPLU1 LU
A71NPLU2 LU
A71NPLU3 LU
A71NPLU4 LU
A71NPLU5 LU
A71NPLU6 LU
A71NPLU7 LU
A71NPLU8 LU
*

* X25.NET DEFINITIONS *

*
RTPNET X25.NET CPHINDX=7,OUHINDX=19,DM=YES,RESETINO=YES,RFAC=FASTSEL, X
 R20=2,R22=2,R23=2,NSTDFAC=NONE, X
 DCI=YES,CCITT=1993
*
X25.VCCPT INDEX=1,MAXPKTL=(4096,4096),VWINDOW=(127,127)
X25.VCCPT INDEX=2,MAXPKTL=(4096,4096),VWINDOW=(127,127)
X25.VCCPT INDEX=3,MAXPKTL=(4096,4096),VWINDOW=(127,127)
X25.VCCPT INDEX=4,MAXPKTL=(4096,4096),VWINDOW=(127,127)
X25.VCCPT INDEX=5,MAXPKTL=(4096,4096),VWINDOW=(127,127)
X25.VCCPT INDEX=6,MAXPKTL=(4096,4096),VWINDOW=(127,127)
X25.VCCPT INDEX=7,MAXPKTL=4096,VWINDOW=127
*
X25.OUFT INDEX=1
X25.OUFT INDEX=2,OPTFACL=420707430202
X25.OUFT INDEX=3,OPTFACL=420707430303,USRFILD=1234567890
X25.OUFT INDEX=4,OPTFACL=420A0A430F0F
*

* PHYSICAL LINE 2304 - X.25 ODLC CONNECTION *

*
MCH2304 X25.MCH ADDRESS=2304,ODLC=YES, X
 RESETPVC=YES, X
 RNRTIMER=0, X
 RNRPKT=NO, X
 FRMLGTH=4100, X

```

Example F—Gen Information for X.25 ODL

```

PKTMODL=128, X
MMODULO=128, X
MWINDOW=127, X
ACCOUNT=YES, X
ANS=CONT, X
DBIT=YES, X
GATE=NO, X
LCN0=NOTUSED, X
LLCLIST=LLC3, X
LSPRI=NO, X
MBITCHN=YES, X
LUNAME=XUA01, X
NCPGRP=XMA01, X
PUNAME=XPA01, X
NDRETRY=1, X
NPRETRY=7, X
PHYSRSC=YES, X
SPEED=2048000, X
STATION=DTE, X
TPTIMER=10, X
XMONLNK=YES, X
NPACOLL=(MCHLINE,MCHPU,VCPU), X
NPADTEAD=712304, * xxaaa xx=subarea aaa=line addr X
SVCINN=2

*
X25.LCG LCGN=0
*

* SUBAREA (INN) PVCs *

*
XL01LPVC X25.LINE DSTNODE=INN,SPAN=OPER1,TYPE=P,MONLINK=CONT, X
NCPGRP=XGA01PVC,LLC=LLC3,LCN=1,VCCINDX=7
XPP01PVC X25.PU ISTATUS=ACTIVE,PUTYPE=4,TGN=1
*
XL02LPVC X25.LINE DSTNODE=INN,SPAN=OPER1,TYPE=P,MONLINK=CONT, X
NCPGRP=XGA01PVC,LLC=LLC3,LCN=2,VCCINDX=7
XPP02PVC X25.PU ISTATUS=ACTIVE,PUTYPE=4,TGN=2
*

* PERIPHERAL (BNN) PVCs *

*
X25.LINE LCN=3,LLC=LLC3,VCCINDX=1,TYPE=P
X25.PU PUTYPE=2,ADDR=01,MAXDATA=265,MAXOUT=1,PACING=1, X
PASSLIM=1,VPACING=(2,1),XID=NO
X25.LU LOCADDR=02,USSTAB=AUSSTAB,LUDR=YES
*

*
X25.LCG LCGN=1
*

* 2 SUBAREA (INN) AND 7 PERIPHERAL (BNN) SVCS *

*
XL01SVC X25.LINE DSTNODE=INN,CALL=INOUT,SPAN=OPER1,TYPE=S
XPP01SVC X25.PU ISTATUS=INACTIVE,PUTYPE=4

```

## Example F—Gen Information for X.25 ODL

```
*
XL02LSVC X25.LINE DSTNODE=INN, CALL=INOUT, SPAN=OPER1, TYPE=S
XPP02SVC X25.PU ISTATUS=INACTIVE, PUTYPE=4
*
 X25.VC LCN=(1,9), TYPE=S, OUFINDX=1, VCCINDX=1, CALL=INOUT, X
 ISTATUS=ACTIVE, HEXNAME=NO, SPAN=OPER1, SUFFIX=1, X
 PRFLINE=XM01LN, PRFPU=XM01PU, PRFLU=XM01LU
*
 X25.END
*

* CHANNEL ADAPTER DEFINITIONS *

*
A71CAG GROUP LNCTL=CA, CA=TYPE7, NCPA=ACTIVE, DELAY=0.0, NPACOLL=NO, X
 MAXBFRU=96, CASDL=420, TIMEOUT=180, ISTATUS=ACTIVE
*
A71C01 LINE ADDRESS=P1, CASDL=420, MONLINK=CONT, DELAY=0.2
A71P01A PU PUTYPE=5, TGN=1
*
A71C03 LINE ADDRESS=P3, CASDL=420, MONLINK=CONT, DELAY=0.2
A71P03A PU PUTYPE=5, TGN=1
*
GENEND GENEND
 END
```



---

## Appendix C. Statement and Keyword Changes

|                               |      |
|-------------------------------|------|
| BUILD                         | C-2  |
| X25.CPL (NPSI V3R4 and Later) | C-3  |
| X25.END                       | C-4  |
| X25.FCG                       | C-4  |
| X25.LCG                       | C-5  |
| X25.LINE                      | C-6  |
| X25.LU                        | C-7  |
| X25.MCH                       | C-8  |
| X25.NET                       | C-11 |
| X25.OUFT                      | C-12 |
| X25.PAD                       | C-12 |
| X25.PU                        | C-12 |
| X25.TRAN                      | C-15 |
| X25.VC                        | C-16 |
| X25.VCCPT                     | C-18 |

This appendix is comprised of tables that list X.25 statements and statement keywords.

All of the keywords for each statement are listed in alphabetical order. The values specified for several of the higher-level definition statements will sift down to the lower-level definition statements if they are not explicitly overridden.

This sifting is indicated in the following tables by placing the lower-level statement name in parentheses. For example, (MCH) is mentioned for MWINDOW in the BUILD statement. This indicates that you can code MWINDOW in the BUILD statement, and that it will sift down to the X25.MCH statement.

**Note:** When *No longer valid* is specified in the disposition, it represents a keyword that was valid for NPSI Version 1 but is not valid for NPSI Version 3 or X.25 ODLIC.

**BUILD**

*Table C-1. BUILD Changes*

| <b>Keyword</b> | <b>Disposition</b>                                 |
|----------------|----------------------------------------------------|
| -----          | Use NCP BUILD                                      |
| BYTTHROD       | Code on NCP BUILD with X25 prefix (V3R4 and later) |
| DDNAME         | No longer valid                                    |
| IDNUMH         | Code on NCP BUILD with X25 prefix                  |
| JOBCARD        | No longer valid                                    |
| MACLIB         | No longer valid                                    |
| MAXPIU         | Code on NCP BUILD with X25 prefix                  |
| MCHCNT         | Code on NCP BUILD with X25 prefix                  |
| MMODULO        | Code on NCP BUILD with X25 prefix (MCH)            |
| MWINDOW        | Code on NCP BUILD with X25 prefix (MCH)            |
| NPA            | Code on NCP BUILD with X25 prefix                  |
| OUTIDNUM       | Code on NCP BUILD with X25 prefix                  |
| OUTPUT         | No longer valid                                    |
| PAHINDEX       | Code on NCP BUILD with X25 prefix                  |
| PREFIX         | Code on NCP BUILD with X25 prefix                  |
| QUALIFY        | No longer valid                                    |
| SESSACC        | Code on NCP BUILD with X25 prefix                  |
| SEGTHROD       | Code on NCP BUILD with X25 prefix                  |
| SNAP           | Code on NCP BUILD with X25 prefix                  |
| SRCHI          | No longer valid                                    |
| SRCLO          | No longer valid                                    |
| SRCPRFX        | No longer valid                                    |
| SSPV2          | No longer valid                                    |
| USGTIER        | Code on NCP BUILD with X25 prefix                  |

---

**X25.CPL (NPSI V3R4 and Later)**


---

*Table C-2 (Page 1 of 2). X25.CPL Changes*


---

| <b>Keyword</b> | <b>Disposition</b>                                                                        |
|----------------|-------------------------------------------------------------------------------------------|
| symbol         | Passed to NCP as is                                                                       |
| ADJLINK        | Passed to NetView (Starting with V3R9)                                                    |
| ADJNODE        | Passed to NetView (Starting with V3R9)                                                    |
| AGGPRI         | Passed to NetView (Starting with V3R9)                                                    |
| ALTCTCP        | Optional                                                                                  |
| ANS            | Passed to NCP as is                                                                       |
| ASLENT         | Passed to VTAM as is                                                                      |
| ASLTAB         | Passed to VTAM as is                                                                      |
| CLRSESSQ       | Passed to VTAM (Starting with V3R9)                                                       |
| CMTKYWRD       | Generates a keyword unknown to NPSI as a comment on an NCP statement (Starting with V3R9) |
| CTCPNO         | Required                                                                                  |
| DISCNT         | Passed to VTAM as is                                                                      |
| DLOGMOD        | Passed to VTAM as is                                                                      |
| ENCR           | Passed to VTAM as is                                                                      |
| FEATUR2        | Passed to VTAM as is                                                                      |
| ISTATUS        | Passed to VTAM as is                                                                      |
| LOGTAB         | Passed to VTAM as is                                                                      |
| LUNAME         | Optional                                                                                  |
| MAXTIME        | Required                                                                                  |
| MDLENT         | Passed to VTAM as is                                                                      |
| MDLTAB         | Passed to VTAM as is                                                                      |
| MINDATA        | Required                                                                                  |
| MODETAB        | Passed to VTAM as is                                                                      |
| NCPGRP         | Optional                                                                                  |
| NGFINC         | Passed to NetView (Starting with V3R9)                                                    |
| NGFOPT         | Passed to NetView (Starting with V3R9)                                                    |
| NGFTXT         | Passed to NetView (Starting with V3R9)                                                    |
| OWNER          | Passed to VTAM as is                                                                      |
| PUNAME         | Optional                                                                                  |
| SPAN           | Passed to NetView                                                                         |
| STATOPT        | Passed to NetView                                                                         |
| SSCPFM         | Passed to VTAM as is                                                                      |
| TERM           | Passed to VTAM as is                                                                      |
| USSTAB         | Passed to VTAM as is                                                                      |
| VPACING        | Passed to VTAM as is                                                                      |

---

## Statement and Keyword Changes

Table C-2 (Page 2 of 2). X25.CPL Changes

| Keyword  | Disposition                                                                  |
|----------|------------------------------------------------------------------------------|
| VTMKYWRD | Generates a keyword unknown to NPSI on an NCP statement (Starting with V3R9) |

## X25.END

Table C-3. X25.END Changes

| Keyword  | Disposition     |
|----------|-----------------|
| symbol   | Optional        |
| INCHI    | No longer valid |
| INCINIT  | No longer valid |
| INCL2HI  | No longer valid |
| INCL2LO  | No longer valid |
| INCPRFX  | No longer valid |
| LSTUACB  | No longer valid |
| NCPSTAG1 | No longer valid |
| ORDHI    | No longer valid |
| ORDINIT  | No longer valid |
| ORDL2HI  | No longer valid |
| ORDL2LO  | No longer valid |
| SRCHI    | No longer valid |
| SRCLO    | No longer valid |
| X25VTAM  | No longer valid |

## X25.FCG

Table C-4 (Page 1 of 2). X25.FCG Changes

| Keyword  | Disposition                                                                               |
|----------|-------------------------------------------------------------------------------------------|
| symbol   | Optional                                                                                  |
| ADJLINK  | Passed to NetView (Starting with V3R9)                                                    |
| ADJNODE  | Passed to NetView (Starting with V3R9)                                                    |
| AGGPRI   | Passed to NetView (Starting with V3R9)                                                    |
| ANS      | Passed to NCP as is                                                                       |
| ASLENT   | Passed to VTAM as is                                                                      |
| ASLTAB   | Passed to VTAM as is                                                                      |
| CLRSESSQ | Passed to VTAM (Starting with V3R9)                                                       |
| CMTKYWRD | Generates a keyword unknown to NPSI as a comment on an NCP statement (Starting with V3R9) |
| CTCPNO   | Required                                                                                  |
| DISCNT   | Passed to VTAM as is                                                                      |

Table C-4 (Page 2 of 2). X25.FCG Changes

| <b>Keyword</b> | <b>Disposition</b>                                                           |
|----------------|------------------------------------------------------------------------------|
| DLOGMOD        | Passed to VTAM as is                                                         |
| ENCR           | Passed to VTAM as is                                                         |
| FEATUR2        | Passed to VTAM as is                                                         |
| HEXNAME        | Optional                                                                     |
| ISTATUS        | Passed to VTAM as is                                                         |
| LOGTAB         | Passed to VTAM as is                                                         |
| MAXDATA        | Passed to NCP as is                                                          |
| MDLENT         | Passed to VTAM as is                                                         |
| MDLTAB         | Passed to VTAM as is                                                         |
| NCPGRP         | Optional                                                                     |
| NGFINC         | Passed to NetView (Starting with V3R9)                                       |
| NGFOPT         | Passed to NetView (Starting with V3R9)                                       |
| NGFTXT         | Passed to NetView (Starting with V3R9)                                       |
| OWNER          | Passed to VTAM as is                                                         |
| PRFLINE        | Optional                                                                     |
| PRFLU          | Optional                                                                     |
| PRFPU          | Optional                                                                     |
| QTY            | Required                                                                     |
| SPAN           | Passed to NetView as is                                                      |
| SSCPFM         | Passed to VTAM as is                                                         |
| STATOPT        | Passed to NetView                                                            |
| SUFFIX         | Optional                                                                     |
| TERM           | Passed to VTAM as is                                                         |
| USSTAB         | Passed to VTAM as is                                                         |
| VPACING        | Optional                                                                     |
| VTMKYWRD       | Generates a keyword unknown to NPSI on an NCP statement (Starting with V3R9) |

**X25.LCG**

Table C-5. X25.LCG Changes

| <b>Keyword</b> | <b>Disposition</b> |
|----------------|--------------------|
| symbol         | Optional           |
| LCGN           | Required           |

## Statement and Keyword Changes

### X25.LINE

Table C-6 (Page 1 of 2). X25.LINE Changes

| Keyword  | Disposition                                                                               |
|----------|-------------------------------------------------------------------------------------------|
| symbol   | Optional                                                                                  |
| ADJLINK  | Passed to NetView (Starting with V3R9)                                                    |
| ADJNODE  | Passed to NetView (Starting with V3R9)                                                    |
| AGGPRI   | Passed to NetView (Starting with V3R9)                                                    |
| CALL     | Optional                                                                                  |
| CMTKYWRD | Generates a keyword unknown to NPSI as a comment on an NCP statement (Starting with V3R9) |
| COMMITO  | Optional                                                                                  |
| CTCPNO   | Optional (V3R4 and later)                                                                 |
| DEFPRI   | Optional (V3R4 and later)                                                                 |
| DSTNODE  | Optional                                                                                  |
| ID       | Optional                                                                                  |
| IDNUMT   | Optional                                                                                  |
| IPL      | Passed to NCP as is                                                                       |
| ISTATUS  | Passed to VTAM as is                                                                      |
| LCN      | Optional                                                                                  |
| LLC      | Required if TYPE=PERMANENT; otherwise it is not valid                                     |
| MODE     | Optional                                                                                  |
| MONLINK  | Passed to NCP as is                                                                       |
| NCPGRP   | Optional                                                                                  |
| NGFINC   | Passed to NetView (V3R8 and Later)                                                        |
| NGFOPT   | Passed to NetView (Starting with V3R9)                                                    |
| NGFTEXT  | Passed to NetView (V3R8 and Later)                                                        |
| NOTOPC5  | Optional                                                                                  |
| OSITYPE  | Optional                                                                                  |
| OUFINDX  | Required if TYPE=SWITCHED; otherwise it is not valid                                      |
| OWNER    | Passed to VTAM as is                                                                      |
| RESETUNB | Optional                                                                                  |
| RETVCCT  | Optional                                                                                  |
| RETVCTO  | Optional                                                                                  |
| SPAN     | Passed to NetView as is                                                                   |
| STATOPT  | Passed to NetView as is                                                                   |
| SWIMMTID | Passed to NCP as is                                                                       |
| TADDR    | Passed to NCP as is                                                                       |
| TRANSFR  | Passed to NCP as is                                                                       |
| TYPE     | Required                                                                                  |

Table C-6 (Page 2 of 2). X25.LINE Changes

| Keyword  | Disposition                                                                  |
|----------|------------------------------------------------------------------------------|
| VCCINDX  | Optional                                                                     |
| VTMKYWRD | Generates a keyword unknown to NPSI on an NCP statement (Starting with V3R9) |

## X25.LU

Table C-7 (Page 1 of 2). X25.LU Changes

| Keyword  | Disposition                                                                               |
|----------|-------------------------------------------------------------------------------------------|
| symbol   | Optional                                                                                  |
| ADJLINK  | Passed to NetView (Starting with V3R9)                                                    |
| ADJNODE  | Passed to NetView (Starting with V3R9)                                                    |
| AGGPRI   | Passed to NetView (Starting with V3R9)                                                    |
| ASLENT   | Passed to VTAM as is                                                                      |
| ASLTAB   | Passed to VTAM as is                                                                      |
| BATCH    | Passed to VTAM as is                                                                      |
| BUFLIM   | Not valid                                                                                 |
| CMTKYWRD | Generates a keyword unknown to NPSI as a comment on an NCP statement (Starting with V3R9) |
| DATASW   | Not valid                                                                                 |
| DLOGMOD  | Passed to VTAM as is                                                                      |
| ENCR     | Passed to VTAM as is                                                                      |
| FEATUR2  | Passed to VTAM as is                                                                      |
| ISTATUS  | Passed to VTAM as is                                                                      |
| LOCADDR  | Passed to NCP as is                                                                       |
| LOGAPPL  | Passed to VTAM as is                                                                      |
| LOGTAB   | Passed to VTAM as is                                                                      |
| LUCB     | Not valid                                                                                 |
| LUDR     | Forced to a value of NO                                                                   |
| LUFVT    | Not valid                                                                                 |
| LUNTFY   | Not valid                                                                                 |
| LUTYPE   | Not valid                                                                                 |
| MAXCOLL  | Not valid                                                                                 |
| MAXSESS  | Passed to VTAM as is                                                                      |
| MDLENT   | Passed to VTAM as is                                                                      |
| MDLTAB   | Passed to VTAM as is                                                                      |
| MODETAB  | Passed to VTAM as is                                                                      |
| NGFOPT   | Passed to NetView                                                                         |
| NETID    | Passed to VTAM as is                                                                      |
| NPACOLL  | Passed to NCP as is                                                                       |

## Statement and Keyword Changes

| <i>Table C-7 (Page 2 of 2). X25.LU Changes</i> |                                                         |                      |
|------------------------------------------------|---------------------------------------------------------|----------------------|
| <b>Keyword</b>                                 | <b>Disposition</b>                                      |                      |
| NUMSESS                                        | Not valid                                               |                      |
| PACING                                         | Passed to NCP as is                                     |                      |
| REGISTER                                       | Passed to NCP as is                                     |                      |
| SPAN                                           | Passed to NetView as is                                 |                      |
| SSCPFM                                         | Passed to VTAM as is                                    |                      |
| STATOPT                                        | Passed to NetView as is                                 |                      |
| TERM                                           | Passed to VTAM as is                                    |                      |
| UCCB                                           | Not valid                                               |                      |
| USSTAB                                         | Passed to VTAM as is                                    |                      |
| VPACING                                        | Passed to VTAM as is                                    |                      |
| VTMKYWRD                                       | Generates a keyword unknown to NPSI on an NCP statement | (Starting with V3R9) |

## X25.MCH

| <i>Table C-8 (Page 1 of 4). X25.MCH Changes</i> |                                                                      |                      |
|-------------------------------------------------|----------------------------------------------------------------------|----------------------|
| <b>Keyword</b>                                  | <b>Disposition</b>                                                   |                      |
| symbol                                          | Optional                                                             |                      |
| ACCOUNT                                         | Optional                                                             | (V3R4 and later)     |
| ACTIVTO                                         | Optional                                                             |                      |
| ADDIFG                                          | Optional                                                             | (V3R8 and later)     |
| ADDRESS                                         | Required                                                             |                      |
| ADJLINK                                         | Passed to NetView                                                    | (Starting with V3R9) |
| ADJNODE                                         | Passed to NetView                                                    | (Starting with V3R9) |
| AEFBYTE                                         | Optional                                                             | (V3R4 and later)     |
| AEFCTCP                                         | Optional                                                             | (V3R4 and later)     |
| AGGPRI                                          | Passed to NetView                                                    | (Starting with V3R9) |
| ANS                                             | Passed to NCP as is                                                  |                      |
| BRKCON                                          | Optional                                                             |                      |
| CAUSE                                           | Optional                                                             |                      |
| CCXDELAY                                        | Optional                                                             |                      |
| CLRSESSQ                                        | Passed to VTAM                                                       | (Starting with V3R9) |
| CMTKYWRD                                        | Generates a keyword unknown to NPSI as a comment on an NCP statement | (Starting with V3R9) |
| CONNECT                                         | Optional                                                             |                      |
| CTCP                                            | Optional                                                             |                      |
| CUD0                                            | Optional                                                             |                      |
| DBIT                                            | Optional                                                             |                      |
| DEFPRI                                          | Optional                                                             | (V3R4 and later)     |



Table C-8 (Page 2 of 4). X25.MCH Changes

| <b>Keyword</b> | <b>Disposition</b>                    |
|----------------|---------------------------------------|
| DIRECT         | Optional                              |
| DSABLTO        | Optional                              |
| DTETODTE       | Optional (X.25 ODLC only)             |
| ENABLTO        | Optional                              |
| FTPI           | Optional (V3R4 and later)             |
| FTPITH         | Optional (Starting with V3R9)         |
| FTPITHTO       | Optional (Starting with V3R9)         |
| FRMLGTH        | Required                              |
| GATE           | Optional                              |
| IDBLKC         | Optional                              |
| IDBLKG         | Optional                              |
| IDBLKLCG       | Optional (V3R8 and Later)             |
| IDBLKP         | Optional                              |
| INNPORT        | Passed to VTAM as is (V3R4 and later) |
| INTFAC         | Optional                              |
| INTWLINK       | Optional (V3R6 and later)             |
| ISTATUS        | Passed to VTAM as is                  |
| ITRACE         | Optional                              |
| LCGDEF         | Required                              |
| LCN0           | Optional                              |
| LLC            | Optional (LINE)                       |
| LLC0           | Optional                              |
| LLC2           | Optional                              |
| LLC3           | Optional                              |
| LLC4           | Optional                              |
| LLC5           | Optional                              |
| LLCI           | Optional                              |
| LLCLIST        | Optional                              |
| LOGAPPL        | Passed to VTAM as is                  |
| LSPRI          | Optional                              |
| LUNAME         | Optional                              |
| MACB           | Optional                              |
| MBITCHN        | Optional                              |
| MBNOFULL       | Optional                              |
| MMODULE        | Optional (V3R7 and later)             |
| MODETAB        | Passed to VTAM as is                  |
| MWINDOW        | Required                              |
| NCPGRP         | Optional                              |

## Statement and Keyword Changes

Table C-8 (Page 3 of 4). X25.MCH Changes

| Keyword  | Disposition                             |
|----------|-----------------------------------------|
| NDRETRY  | Optional                                |
| NETID    | Optional                                |
| NGFINC   | Passed to NetView (V3R8 and Later)      |
| NGFOPT   | Passed to NetView (Starting with V3R9)  |
| NGFTEXT  | Passed to NetView (V3R8 and Later)      |
| NPADTEAD | (V3R4 and Later)                        |
| NPPVCN   | Optional (V3R4 and later)               |
| NPRETRY  | Optional                                |
| ODLC     | Optional (X.25 ODLC and Later)          |
| OWNER    | <i>owner_name</i> values passed to VTAM |
| PAD      | Optional                                |
| PADBRKCD | Optional                                |
| PADCHECK | Optional                                |
| PADINDX  | Optional                                |
| PHYSRSC  | Optional (V3R5 and later)               |
| PKTMODL  | Optional                                |
| PLPIGGYB | Optional                                |
| PUNAME   | Optional                                |
| PWPROT   | Optional                                |
| RESETPVC | Optional (V3R7 and later)               |
| RESETUNB | Optional (Starting with V3R9)           |
| ROLE     | Passed to NCP as is                     |
| RNRTIMER | Optional (V3R7 and later)               |
| RPADPARM | Optional (V3R5 and later)               |
| SDRTCNT  | Optional                                |
| SDRTIME  | Optional                                |
| SHM      | Optional                                |
| SHUTD    | Optional                                |
| SPAN     | Passed to NetView as is                 |
| SPEED    | Optional                                |
| SPNQLLC  | Optional                                |
| STATION  | Optional                                |
| STATOPT  | Passed to NetView as is                 |
| SUBADDR  | Optional                                |
| SUBD     | Optional                                |
| SVCINN   | Optional                                |
| TAXUNIT  | Optional                                |
| TDTIMER  | Optional                                |

Table C-8 (Page 4 of 4). X25.MCH Changes

| <b>Keyword</b> | <b>Disposition</b>                                      |                      |
|----------------|---------------------------------------------------------|----------------------|
| TPTIMER        | Optional                                                |                      |
| T1TIMER        | Optional                                                |                      |
| TRAN           | Optional                                                |                      |
| TRANSFR        | Optional                                                | (V3R5 and later)     |
| T3             | Optional                                                | (V3R4 and later)     |
| T4             | Optional                                                |                      |
| VCID           | Optional                                                |                      |
| VMODTAB        | Passed to VTAM as is                                    |                      |
| VTMKYWRD       | Generates a keyword unknown to NPSI on an NCP statement | (Starting with V3R9) |
| XMONLNK        | Optional                                                |                      |
| X21INACT       | Optional                                                |                      |
| X21NTWK        | Optional                                                |                      |
| X21RTYCT       | Optional                                                |                      |
| X21RTYTO       | Optional                                                |                      |
| X21SW          | Optional                                                |                      |

## X25.NET

Table C-9 (Page 1 of 2). X25.NET Changes

| <b>Keyword</b> | <b>Disposition</b> |                  |
|----------------|--------------------|------------------|
| symbol         | Optional           |                  |
| CAUSE          | Optional           |                  |
| CCITT          | Optional           | (X.25 ODLC only) |
| CPHINDX        | Optional           |                  |
| DCI            | Optional           | (V3R6 and later) |
| DDXP           | No longer valid    |                  |
| DM             | Required           |                  |
| DSTNODE        | Optional           | (LINE)           |
| FRMLGTH        | Optional           | (MCH)            |
| INACTO         | Optional           |                  |
| LLC            | Optional           | (LINE)           |
| LLCLIST        | Optional           | (MCH)            |
| MWINDOW        | Optional           | (MCH)            |
| NETTYPE        | Optional           |                  |
| NSTDFAC        | Optional           |                  |
| OUHINDX        | Optional           |                  |
| PKTMODL        | Optional           | (MCH)            |
| RESETINO       | Optional           |                  |

## Statement and Keyword Changes

Table C-9 (Page 2 of 2). X25.NET Changes

| Keyword  | Disposition                   |
|----------|-------------------------------|
| RESETPVC | Optional (V3R7 and later)     |
| RESETUNB | Optional (Starting with V3R9) |
| RFAC     | Optional                      |
| RNRTIMER | Optional (V3R7 and later)     |
| R20      | Optional                      |
| R22      | Optional                      |
| R23      | Optional                      |
| STATION  | Optional (MCH)                |
| T20      | Optional                      |
| T21      | Optional                      |
| T22      | Optional                      |
| T23      | Optional                      |
| T25      | Optional                      |
| T26      | Optional                      |

## X25.OUFT

Table C-10. X25.OUFT Changes

| Keyword | Disposition |
|---------|-------------|
| symbol  | Optional    |
| INDEX   | Required    |
| OPTFACL | Optional    |
| USRFILD | Optional    |
| USRFIL2 | Optional    |

## X25.PAD

Table C-11. X25.PAD Changes

| Keyword | Disposition |
|---------|-------------|
| symbol  | Optional    |
| INDEX   | Required    |
| PADPARM | Required    |

## X25.PU

Table C-12 (Page 1 of 4). X25.PU Changes

| Keyword | Disposition |
|---------|-------------|
| symbol  | Optional    |

Table C-12 (Page 2 of 4). X25.PU Changes

| <b>Keyword</b> | <b>Disposition</b>                                                   |                                    |
|----------------|----------------------------------------------------------------------|------------------------------------|
| ADDR           | Passed to NCP as is                                                  |                                    |
| ADJLINK        | Passed to NetView                                                    | (Starting with V3R9)               |
| ADJNODE        | Passed to NetView                                                    | (Starting with V3R9)               |
| AGGPRI         | Passed to NetView                                                    | (Starting with V3R9)               |
| ANS            | Passed to NCP as is                                                  |                                    |
| ASLENT         | Passed to VTAM as is                                                 |                                    |
| ASLTAB         | Passed to VTAM as is                                                 |                                    |
| AVGPB          | Passed to NCP as is                                                  |                                    |
| BATCH          | No longer valid                                                      |                                    |
| BRKCON         | Passed to NCP as is                                                  |                                    |
| BUFLIM         | Not valid                                                            |                                    |
| BNNSUP         | Passed to NCP as is                                                  |                                    |
| CAPACITY       | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| CLRSESSQ       | Passed to VTAM                                                       | (Starting with V3R9)               |
| CMTKYWRD       | Generates a keyword unknown to NPSI as a comment on an NCP statement | (Starting with V3R9)               |
| CONNTYPE       | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| COSTBYTE       | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| COSTTIME       | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| CPCP           | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| CPNAME         | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| DATMODE        | Passed to NCP as is                                                  |                                    |
| DISCNT         | Passed to VTAM as is                                                 |                                    |
| DLOGMOD        | Passed to VTAM as is                                                 |                                    |
| DYNADJCP       | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| DYNPU          | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| DYNPUPFX       | Passed to VTAM as is                                                 | (V3R6 and later)                   |
| ENCR           | Passed to VTAM as is                                                 |                                    |
| FEATUR2        | Passed to VTAM as is                                                 |                                    |
| HPR            | Passed to VTAM as is                                                 | (X.25 ODLC starting with NCP V7R5) |
| HPRQLIM        | Optional                                                             | (X.25 ODLC starting with NCP V7R5) |
| IRETRY         | Passed to NCP as is                                                  |                                    |
| ISTATUS        | Passed to VTAM as is                                                 |                                    |
| LLERP          | Required if using HPR over X.25 ODLC                                 | (X.25 ODLC starting with NCP V7R5) |
| LOGAPPL        | Passed to VTAM as is                                                 |                                    |
| LOGTAB         | Passed to VTAM as is                                                 |                                    |

## Statement and Keyword Changes

| <i>Table C-12 (Page 3 of 4). X25.PU Changes</i> |                                         |
|-------------------------------------------------|-----------------------------------------|
| <b>Keyword</b>                                  | <b>Disposition</b>                      |
| LPDA                                            | Not valid                               |
| LUCB                                            | Not valid                               |
| LUFVT                                           | Not valid                               |
| LUNTFY                                          | Not valid                               |
| LUTYPE                                          | Not valid                               |
| MAXDATA                                         | Passed to NCP as is                     |
| MAXLU                                           | Passed to NCP as is                     |
| MAXOUT                                          | Passed to NCP as is                     |
| MDLENT                                          | Passed to VTAM as is                    |
| MDLTAB                                          | Passed to VTAM as is                    |
| MODETAB                                         | Passed to VTAM as is                    |
| NGFINC                                          | Passed to NetView (Starting with V3R9)  |
| NGFOPT                                          | Passed to NetView (Starting with V3R9)  |
| NGFTXT                                          | Passed to NetView (Starting with V3R9)  |
| NETID                                           | Passed to NCP as is                     |
| NN                                              | Passed to NCP as is (V3R6 and later)    |
| PACING                                          | Passed to NCP as is                     |
| PASSLIM                                         | Passed to NCP as is                     |
| PDELAY                                          | Passed to NCP as is (V3R6 and later)    |
| PUCB                                            | Not valid                               |
| PUDR                                            | Forced to a value of NO                 |
| PUFVT                                           | Not valid                               |
| PUNTFY                                          | Not valid                               |
| PUTYPE                                          | Passed to NCP as is                     |
| RETRIES                                         | Passed to NCP as is                     |
| SECLEVEL                                        | Passed to VTAM as is (V3R5 and earlier) |
| SECNET                                          | Passed to VTAM as is                    |
| SECURITY                                        | Passed to VTAM as is (V3R6 and later)   |
| SPAN                                            | Passed to NetView as is                 |
| SRT                                             | Not valid                               |
| SSCPFM                                          | Passed to VTAM as is                    |
| STATOPT                                         | Passed to VTAM as is                    |
| SUBAREA                                         | Passed to NCP as is                     |
| TERM                                            | Passed to VTAM as is                    |
| TGN                                             | Passed to NCP as is                     |
| TGP                                             | Passed to VTAM as is (V3R6 and later)   |
| TOPO                                            | Passed to VTAM as is (V3R6 and later)   |
| UPARM1                                          | Passed to VTAM as is (V3R6 and later)   |

Table C-12 (Page 4 of 4). X25.PU Changes

| <b>Keyword</b> | <b>Disposition</b>                                                           |
|----------------|------------------------------------------------------------------------------|
| UPARM2         | Passed to VTAM as is (V3R6 and later)                                        |
| UPARM3         | Passed to VTAM as is (V3R6 and later)                                        |
| USSTAB         | Passed to VTAM as is                                                         |
| VERALSID       | Passed to VTAM as is (V3R6 and later)                                        |
| VPACING        | Passed to VTAM as is                                                         |
| VTMKYWRD       | Generates a keyword unknown to NPSI on an NCP statement (Starting with V3R9) |
| XID            | Passed to NCP as is                                                          |

## X25.TRAN

Table C-13 (Page 1 of 2). X25.TRAN Changes

| <b>Keyword</b> | <b>Disposition</b> |
|----------------|--------------------|
| symbol         | Optional           |
| DCIN0          | Required           |
| DCIN1          | Required           |
| DCIN2          | Required           |
| DCIN3          | Required           |
| DCIN4          | Required           |
| DCIN5          | Required           |
| DCIN6          | Required           |
| DCIN7          | Required           |
| DCIN8          | Required           |
| DCIN9          | Required           |
| DCINA          | Required           |
| DCINB          | Required           |
| DCINC          | Required           |
| DCIND          | Required           |
| DCINE          | Required           |
| DCINF          | Required           |
| DCOT0          | Required           |
| DCOT1          | Required           |
| DCOT2          | Required           |
| DCOT3          | Required           |
| DCOT4          | Required           |
| DCOT5          | Required           |
| DCOT6          | Required           |
| DCOT7          | Required           |
| DCOT8          | Required           |

## Statement and Keyword Changes

Table C-13 (Page 2 of 2). X25.TRAN Changes

| <b>Keyword</b> | <b>Disposition</b> |
|----------------|--------------------|
| DCOT9          | Required           |
| DCOTA          | Required           |
| DCOTB          | Required           |
| DCOTC          | Required           |
| DCOTD          | Required           |
| DCOTE          | Required           |
| DCOTF          | Required           |
| USER           | Required           |

## X25.VC

Table C-14 (Page 1 of 3). X25.VC Changes

| <b>Keyword</b> | <b>Disposition</b>                                                                        |
|----------------|-------------------------------------------------------------------------------------------|
| symbol         | Optional                                                                                  |
| ADJLINK        | Passed to NetView (Starting with V3R9)                                                    |
| ADJNODE        | Passed to NetView (Starting with V3R9)                                                    |
| AGGPRI         | Passed to NetView (Starting with V3R9)                                                    |
| ANS            | Passed to NCP as is                                                                       |
| ASLENT         | Passed to VTAM as is                                                                      |
| ASLTAB         | Passed to VTAM as is                                                                      |
| CALL           | Optional                                                                                  |
| CAPACITY       | Passed to VTAM as is (V3R6)                                                               |
| CLRSESSQ       | Passed to VTAM (Starting with V3R9)                                                       |
| CMTKYWRD       | Generates a keyword unknown to NPSI as a comment on an NCP statement (Starting with V3R9) |
| COMMITO        | Optional                                                                                  |
| CONNTYPE       | Passed to VTAM as is (V3R6 and later)                                                     |
| COSTBYTE       | Passed to VTAM as is (V3R6 and later)                                                     |
| COSTTIME       | Passed to VTAM as is (V3R6 and later)                                                     |
| CPCP           | Passed to VTAM as is (V3R6 and later)                                                     |
| CPNAME         | Passed to VTAM as is (V3R6 and later)                                                     |
| CTCPNO         | Optional (V3R4 and later)                                                                 |
| DEFPRI         | Optional (V3R4 and later)                                                                 |
| DISCNT         | Passed to VTAM as is                                                                      |
| DLOGMOD        | Passed to VTAM as is                                                                      |
| DYNADJCP       | Passed to VTAM as is (V3R6 and later)                                                     |
| DYNPU          | Passed to VTAM as is (V3R6 and later)                                                     |
| DYNPUPFX       | Passed to VTAM as is (V3R6 and later)                                                     |
| ENCR           | Passed to VTAM as is                                                                      |



Table C-14 (Page 2 of 3). X25.VC Changes

| <b>Keyword</b> | <b>Disposition</b>                                    |                                    |
|----------------|-------------------------------------------------------|------------------------------------|
| FEATUR2        | Passed to VTAM as is                                  |                                    |
| HEXNAME        | Optional                                              |                                    |
| HPR            | Passed to VTAM as is                                  | (X.25 ODLC starting with NCP V7R5) |
| HPRQLIM        | Optional                                              | (X.25 ODLC starting with NCP V7R5) |
| ID             | Optional                                              |                                    |
| IDNUMT         | Optional                                              |                                    |
| ISTATUS        | Passed to VTAM as is                                  |                                    |
| LCN            | Required                                              |                                    |
| LLC            | Required if TYPE=PERMANENT; otherwise it is not valid |                                    |
| LLERP          | Required if using HPR over X.25 ODLC                  | (X.25 ODLC starting with NCP V7R5) |
| LOGAPPL        | Passed to VTAM as is                                  |                                    |
| LOGTAB         | Passed to VTAM as is                                  |                                    |
| MAXDATA        | Passed to NCP as is                                   |                                    |
| MAXOUT         | Optional                                              |                                    |
| MDLENT         | Passed to VTAM as is                                  |                                    |
| MDLTAB         | Passed to VTAM as is                                  |                                    |
| MODETAB        | Passed to VTAM as is                                  |                                    |
| NCPGRP         | Optional                                              |                                    |
| NGFINC         | Passed to NetView                                     | (V3R8 and Later)                   |
| NGFOPT         | Passed to NetView                                     | (Starting with V3R9)               |
| NGFTEXT        | Passed to NetView                                     | (V3R8 and Later)                   |
| NN             | Passed to NCP as is                                   | (V3R6 and later)                   |
| OSITYPE        | Optional                                              |                                    |
| OUFINDX        | Required if TYPE=SWITCHED; otherwise it is not valid  |                                    |
| OWNER          | Passed to VTAM as is                                  |                                    |
| PDELAY         | Passed to NCP as is                                   | (V3R6 and later)                   |
| PRFLINE        | Optional                                              |                                    |
| PRFLU          | Optional                                              |                                    |
| PRFPU          | Optional                                              |                                    |
| RESETUNB       | Optional                                              | (Starting with V3R9)               |
| RETVCCCT       | Optional                                              |                                    |
| RETVCTO        | Optional                                              |                                    |
| SECLEVEL       | Passed to VTAM as is                                  | (V3R5 and earlier)                 |
| SECURITY       | Passed to VTAM as is                                  | (V3R6 and later)                   |
| SPAN           | Passed to NetView as is                               |                                    |

## Statement and Keyword Changes

Table C-14 (Page 3 of 3). X25.VC Changes

| <b>Keyword</b> | <b>Disposition</b>                                                           |
|----------------|------------------------------------------------------------------------------|
| SSCPFM         | Passed to VTAM as is                                                         |
| STATOPT        | Passed to NetView as is                                                      |
| SUFFIX         | Optional                                                                     |
| TERM           | Passed to VTAM as is                                                         |
| TGN            | Passed to VTAM as is                                                         |
| TGP            | Passed to VTAM as is (V3R6 and later)                                        |
| TOPO           | Passed to VTAM as is (V3R6 and later)                                        |
| TYPE           | Required                                                                     |
| UPARM1         | Passed to VTAM as is (V3R6 and later)                                        |
| UPARM2         | Passed to VTAM as is (V3R6 and later)                                        |
| UPARM3         | Passed to VTAM as is (V3R6 and later)                                        |
| USSTAB         | Passed to VTAM as is                                                         |
| VCCINDX        | Required                                                                     |
| VERALSID       | Passed to VTAM as is (V3R6 and later)                                        |
| VPACING        | Passed to VTAM as is                                                         |
| VTMKYWRD       | Generates a keyword unknown to NPSI on an NCP statement (Starting with V3R9) |

## X25.VCCPT

Table C-15. X25.VCCPT Changes

| <b>Keyword</b> | <b>Disposition</b> |
|----------------|--------------------|
| symbol         | Optional           |
| INDEX          | Required           |
| INSLOW         | Optional           |
| MAXPKTL        | Required           |
| VWINDOW        | Optional           |

---

## Appendix D. Macros Shipped with NPSI

The macros identified in this appendix are provided as programming interfaces for customers by NPSI.

**Note:** Do not use any NPSI macros as programming interfaces other than those identified in this appendix.

The following macros are general-use programming interfaces:

- BAKACUDA
- BAKAVT
- BAKAVUDA
- BAKAXB2
- BAKCINOP
- BAKCOMAT
- BAKGLOB
- BAKISB
- BAKLIQ2
- BAKLLCT
- BAKLUAT
- BAKMKBAT
- BAKMKB2
- BAKMKX
- BAKMUA2
- BAKMX21
- BAKOCUD0
- BAKOUFT
- BAKOUTIT
- BAKPADP
- BAKPADPT
- BAKPML
- BAKPMP
- BAKPVP
- BAKQCB
- BAKRFAC
- BAKSLUB
- BAKSOT
- BAKTMR
- BAKUST1 (shipped on base tape)
- BAKUST2 (shipped on base tape) Usage Tier 2
- BAKUST25 (shipped on base tape) Usage Tier 3
- BAKUST3 (shipped on base tape) Usage Tier 3
- BAKUST45 (shipped on base tape) Usage Tier 4
- BAKUST5 (shipped on base tape) Usage Tier 5
- BAKVCB
- BAKVCBAT
- BAKVCCPT
- BAKVCGAT
- BAKVCT
- BAKVCX
- BAKVPTIT
- BAKVUALK

## Macros

- BAKVUA2
- BAKXUA2
- BAKX25EQ
- BAK2CPL
- BAK2END
- BAK2INIT
- BAK2LCG
- BAK2MC2
- BAK2NET
- BAK2OUFT
- BAK2PAPT
- BAK2TRAN
- BAK2VC
- BAK2VCPT
- KDC
- KDEFMSK
- KEQU

---

## Appendix E. NPSI Error Messages

This appendix contains all BAL error messages generated for X.25..

---

**BAL001I: aaa = bbb INVALID, REQUIRED**

**Explanation:** bbb is not a valid specification for keyword aaa.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL002I: aaa = bbb INVALID, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL003I: aaa = bbb INVALID, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa.

**Utility Action:** Operand aaa is not required and the value bbb is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL004I: aaa = bbb INVALID, NOT WITHIN RANGE ccc, REQUIRED**

**Explanation:** The value bbb specified for keyword aaa is not within the valid range of values ccc.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL005I: aaa = bbb INVALID, NOT WITHIN RANGE ccc, ddd ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The value bbb specified for keyword aaa is not within the valid range of values ccc.

**Utility Action:** The default value ddd is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL006I: FTPI FUNCTION NOT SUPPORTED BY AN X.25 MCH WHILE THERE IS A X25.CPL DEFINITION CODED.**

**Explanation:** If X25.CPL is coded, at least one MCH must be dedicated to FTPI.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL007I: aaa = bbb INVALID, ONLY n SUBOPERAND(S) ALLOWED, REQUIRED**

**Explanation:** bbb is not a valid specification for keyword aaa. You can code only n subvalues.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL008I: aaa = bbb INVALID, ONLY n SUBOPERAND(S) ALLOWED, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. You can code only n subvalues.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL009I: aaa = bbb INVALID, ONLY n SUBOPERAND(S) ALLOWED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. You can code only n subvalues.

**Utility Action:** Operand aaa is not required and the value bbb is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL010I: aaa = bbb INVALID, NOT DEFINED ON ccc, REQUIRED**

**Explanation:** bbb is not a valid specification for keyword aaa. You must also code aaa = bbb on statement ccc.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL011I: aaa NOT SPECIFIED, REQUIRED**

**Explanation:** Operand aaa is not specified.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL012I: aaa OMITTED, REQUIRED WHEN bbb SPECIFIED**

**Explanation:** Operand aaa is not specified. It is required if you code keyword bbb.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL013I: PARAMETERS CONFLICT, aaa = bbb INVALID WHEN ccc SPECIFIED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa when you code the ccc keyword.

**Utility Action:** Operand aaa is not required and the value bbb is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL014I: bbb, xxx, ADDED TO aaa.**

**Explanation:** The keyword value bbb has been added to the keyword aaa for the reason explained by the xxx field. (Example: If LLC3 is not coded in LLCLIST while the SVCSC function is defined on the MCH, LLC3 is added to LLCLIST by the generation process.)

**Utility Action:** The generation process continues to validate the generation definition.

**Programmer Action:** This is an informational message.

---

**BAL015I: PARAMETERS CONFLICT, aaa INVALID WHEN bbb SPECIFIED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** There is a conflict between keyword aaa and keyword bbb.

**Utility Action:** Operand aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL016I: NEWDEFN=YES NOT SPECIFIED ON FIRST OPTIONS STATEMENT, REQUIRED FOR AN X.25 NCP PACKET SWITCHING INTERFACE (NPSI) GENERATION**

**Explanation:** NEWDEFN=YES is not coded on the OPTIONS statement

**Utility Action:** NEWDEFN=YES is required for NPSI. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL017I: PARAMETERS CONFLICT, aaa KEYWORD ONLY ALLOWED IF bbb IS CODED ON ccc, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** Keyword aaa is valid only if bbb keyword is coded on ccc statement.

**Utility Action:** Operand aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL018I: PARAMETERS CONFLICT, aaa = bbb INVALID WHEN ccc SPECIFIED, ddd ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** There is a conflict between keyword aaa = bbb and keyword ccc.

**Utility Action:** The default value ddd is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

---

**BAL019I: aaa = bbb INVALID, aaa MUST BE GREATER THAN ccc, ddd ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. The value of keyword aaa must be greater than the value of keyword ccc.

**Utility Action:** The default value ddd is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL020I: aaa = bbb MUST BE LESS THAN OR EQUAL TO ccc CODED ON ddd, REQUIRED**

**Explanation:** bbb is not a valid specification for keyword aaa. The value of keyword aaa must be less than or equal to the value of keyword ccc coded on the ddd statement.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL021I: PARAMETERS CONFLICT, NPSI VxRy DOES NOT SUPPORT THE NCP VERSION OR THE MODEL CODED**

**Explanation:** NPSI Version x Release y does not support the NCP VERSION or the MODEL values coded on the BUILD statement.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL022I: aaa INVALID, UNEVEN NUMBER OF DIGITS, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** Operand aaa must have a string length of even parity.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL023I: aaa INVALID, INVALID NUMBER OF DIGITS, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The number of characters coded for keyword aaa exceeds the maximum number of characters allowed for the keyword. For the X25.OUFT statement, the values coded for the USRFIL1 and USRFIL2 keywords cannot exceed 254. The sum of the values cannot exceed 256.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL024I: aaa = bbb INVALID, 1 (OR 2) DIGIT(S) REQUIRED IF CODED, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. The string length must be of 1 or 2 characters.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.



---

**BAL025I: aaa = bbb INVALID, FORMAT ERROR, LCGDEF MUST BE CODED IN THE FORM LCGDEF=(LCG,LCNHI), REQUIRED**

**Explanation:** The LCGDEF keyword coded on X25.MCH must have the following format:

LCGDEF=(l cg,l cnhi) or ((l cg,l cnhi),(l cg,l cnhi),...).

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL026I: aaa = bbb INVALID, TOO MANY DIGITS PAST DECIMAL POINT, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. Too many digits coded past the decimal point.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL027I: aaa = bbb INVALID, INVALID DIGIT IN NUMBER, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. A character that is not valid has been coded in the decimal number bbb.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL028I: VCCINDX POINTS TO AN INVALID VCCPT ENTRY, aaa (CODED ON X25.VCCPT) MUST BE LESS THAN bbb (CODED ON X25.MCH), REQUIRED**

**Explanation:** The VCCINDX keyword coded on X25.LINE or X25.VC specifies the entry in the connection parameter table that contains the operational characteristics of the virtual circuit defined by this X25.LINE or X25.VC. The entry in the connection parameter table is selected by the INDEX keyword of a X25.VCCPT statement. A checking is done between the characteristics of that entry and the characteristics defined on the X25.MCH statement.

- The VWINDOW value (coded on X25.VCCPT) must be less than the packet modulo defined in the X25.MCH (PKTMODL).
- The maximum length of the packets to be sent or received over the virtual circuit must be less than the maximum frame length that flows over the physical circuit (FRMLGTH on X25.MCH).

**Note:** The maximum packet length is the sum of the data packet (MAXPKTL in X25.VCCPT) and the packet header; 3 bytes for packet modulo 8, and 4 bytes for packet modulo 128.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL029I: aaa = bbb INVALID, INVALID NUMBER OF SUBOPERANDS, n EXPECTED, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid specification for keyword aaa. The number of subkeywords that was coded is not valid; n subkeywords are expected.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL030I: aaa = n INVALID, NOT DEFINED ON INDEX(bbb), IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** Operand aaa = n defines an entry in an indexed table. One statement is required for each entry in that table. Therefore the entry n has not been coded on the *index* keyword of the bbb statement. (The *index* keyword can be the VCCINDX on the X25.VCCPT statement, the OUFINDX keyword on the X25.OUFT statement, the PADINDX on the X25.PAD statement,...).

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL031I: aaa [(x)] = bbb INVALID, DUPLICATES NOT ALLOWED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The value bbb specified for keyword aaa is not valid. If (x) is present, it shows the subkeyword position of bbb within keyword aaa.

For example, this message applies to:

- NCPGRP—Duplicate GROUP name is not allowed.
- RFAC—Do not repeat the mnemonic code of the facility to be rejected in the sublist of rejected facilities.
- LLCLIST—You must define each LLC type only once in the sublist of LLC type for switched virtual circuit.
- ACCOUNT—You must define each LLC type only once in the sublist of LLC type on which NPM accounting is requested.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL032I: LLC0, LLC2, LLC3, LLC4 AND LLC5 MUST ALL BE UNIQUE, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** When you define SUBADDRESSING to select the LLC type, the definition of subaddressing digits must be unique for each LLC type. Example: X25.MCH SUBADDR=YES, LLC0=(1,2), LLC2=(0,3),...

**Utility Action:** The duplicated keyword is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL033I: Reserved.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

---

**BAL034I: Reserved.**

**Explanation:** Not applicable.  
**Utility Action:** Not applicable.  
**Programmer Action:** Not applicable.

---

**BAL035I: ONE OF LLC0,LLC2, LLC3, LLC4 OR LLC5 KEYWORD MUST BE SPECIFIED IF SUBADDR=YES AND GATE=GENERAL, REQUIRED**

**Explanation:** When you define SUBADDRESSING to select the LLC type, at least one LLCx keyword must be defined. Example: X25.MCH SUBADDR=YES, LLC0=(1,2).  
**Utility Action:** One LLCx keyword is required. The generation process ends after validation of the generation definition.  
**Programmer Action:** Correct the error and resubmit the job.

---

**BAL036I: PARAMETERS CONFLICT, PKTMODL = n INVALID WHEN FRMLGTH = m, DUE TO PACKET HEADER LENGTH, REQUIRED**

**Explanation:** There is a mismatch between the packet modulo and the maximum frame length specified on X25.MCH (PKTMODL=n, FRMLGTH=m). PKTMODL=8 matches with FRMLGTH in the range 35–4099. PKTMODL=128 matches with FRMLGTH in the range 36–4100.  
**Utility Action:** Valid correspondence between PKTMODL and FRMLGTH is required. The generation process ends after validation of the generation definition.  
**Programmer Action:** Correct the error and resubmit the job.

---

**BAL037I: THE NUMBER OF X25.MCHS CODED IS GREATER THAN THE NUMBER SPECIFIED ON X25.MCHCNT (BUILD)**

**Explanation:** The X25.MCHCNT keyword must indicate the number of physical circuits defined in the entire NPSI generation (number of X25.MCH).  
**Utility Action:** The X25.MCHCNT keyword is required. The generation process ends after validation of the generation definition.  
**Programmer Action:** Correct the error and resubmit the job.

---

**BAL038I: aaa MUST BE CODED IN ASCENDING ORDER, REQUIRED**

**Explanation:** The value coded for keyword aaa (LCGN number, LCN number,...) must be coded in ascending order in the following statement (X25.LCGN, X25.VC, X25.LINE,...).  
**Utility Action:** Ascending order for keyword aaa is required. The generation process ends after validation of the generation definition.  
**Programmer Action:** Correct the error and resubmit the job.

---

**BAL039I: SEQUENCE ERROR, aaa OUT OF ORDER, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The statement aaa is misplaced.  
**Utility Action:** The statement aaa is ignored. The generation process ends after validation of the generation definition.  
**Programmer Action:** Correct the error and resubmit the job.

---

**BAL040I: aaa NO LONGER SUPPORTED, IGNORED**

**Explanation:** The keyword aaa is not available for the current level of NPSI.

**Utility Action:** The keyword aaa is ignored. This message is only an informational message, the generation process is not interrupted.

**Programmer Action:** You can remove these keywords from your generation definition.

---

**BAL041I: aaa = bbb INVALID, TOO MANY CHARACTERS, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** bbb is not a valid value for keyword aaa. Too many characters coded.

**Utility Action:** The default value ccc is assumed for keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL042I: aaa = bbb INVALID, NOT WITHIN RANGE n to m, REQUIRED**

**Explanation:** bbb is not a valid value for keyword aaa. The keyword value must be in the range n to m.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL043I: aaa = bbb INVALID, REQUIRED**

**Explanation:** bbb is not a valid value for keyword aaa.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL044I: aaa [(x)] = bbb INVALID, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The value bbb specified for keyword aaa is not valid. If (x) is present, it shows the subkeyword position of bbb within keyword aaa.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL045I: LCGDEF(x)=aaa INVALID, ALL LCGDEF(LCG) MUST BE UNIQUE, REQUIRED**

**Explanation:** x shows the position of the pair (lcg,lcghi) within the keyword LCGDEF. Each logical channel group (LCG) can be defined only once.

**Utility Action:** The LCGDEF keyword is required on X25.MCH. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL046I: aaa INVALID, ONLY n aaa STATEMENT(S) ALLOWED IN AN X.25 NCP PACKET SWITCHING INTERFACE (NPSI) GENERATION, IGNORED**

**Explanation:** Statement aaa is not valid. Only n aaa statements can be coded in the entire NPSI generation (up to nine X25.NET, up to two X25.TRAN,...).

**Utility Action:** Statement aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL047I: {INDEX] USER}=n INVALID, PREVIOUSLY DEFINED, IGNORED FOR STATEMENT USER KEYWORD VALIDATION**

**Explanation:** 1-INDEX=n applies to X25.PAD, X25.OUFT, X25.VCCPT. INDEX=n defines an entry in the appropriate table and cannot be referenced twice. 1-USER=n applies to X25.TRAN. USER=n specifies the translate table; only two user translate tables can be generated for NPSI; the same table cannot be generated twice.

**Utility Action:** The statement is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL048I: PARAMETERS CONFLICT, aaa INVALID IF bbb CODED (OR DEFAULT) ON ccc, REQUIRED**

**Explanation:** aaa keyword is not valid if the bbb keyword is coded or defaulted on the ccc keyword or statement

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL049I: NO MORE IDENTIFICATION GROUP AVAILABLE**

**Explanation:** A label is attached to each NCP GROUP that includes the virtual circuits. This is done by either the user, when specifying NCPGRP=NEW, or a label name (such as X25.LINE or X25.VC), or generated by NPSI. If the user specifies NCPGRP=NEW, or if NPSI generates a label, the following default name is built:

p25Pxxxy for PVCs  
p25Sxxxy for SVCs

where:

p is the X25.PREFIX keyword of the BUILD statement  
xxx is the physical circuit line address as coded in the X25.MCH keyword  
y is the character identifying the group sequence on the physical circuit.

39 characters can be assigned as identification group:

letters 'a' to 'b'  
number '0' to '9'  
characters @, #, \$

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Reduce the number of NCP GROUP names generated by NPSI and resubmit the job.

---

**BAL050I: X25.MCHCNT(BUILD) CODED BUT NO X25.MCH NPSI STATEMENT PROCESSED DURING THE GENERATION**

**Explanation:** In a NPSI generation deck, the number of physical circuits defined by X25.MCHCNT keyword must match with the number of X25.MCH statements coded.

**Utility Action:** As many X25.MCH statements as the number specified on the X25.MCHCNT keyword must be coded. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL051I: PARAMETERS CONFLICT, aaa INVALID IF bbb OMITTED ON ccc, REQUIRED**

**Explanation:** aaa keyword is not valid if the bbb keyword is omitted on the ccc statement.

**Utility Action:** The association of aaa and bbb keywords is required for statement validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Ensure that the number of X25.MCH statements coded equals the value coded for the X25.MCHCNT parameter of the BUILD statement. Correct the error and resubmit the job.

---

**BAL052I: PARAMETERS CONFLICT, aaa = bbb INVALID WITH ccc STATEMENT REQUIRED**

**Explanation:** aaa = bbb keyword is not allowed to be coded on the ccc statement.

**Utility Action:** Remove the aaa keyword from the ccc statement, required for statement validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL053I: LCN = bbb INVALID, ONLY 1 OR 2 SUBOPERANDS ALLOWED ON LCN KEYWORD OF X25.VC STATEMENT**

**Explanation:** Too many subkeywords have been coded on the LCN keyword for X25.VC statement; two subkeywords maximum are allowed. Example: LCN=(1rst lcn,last lcn)

**Utility Action:** Verify the number of virtual circuits to be defined by this X25.VC. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL054I: LCN = aaa INVALID, THE LCN VALUE CODED IS GREATER THAN THE HIGHER VALUE CODED IN X25.MCH(LCGDEF), REQUIRED**

**Explanation:** The LCN value coded is greater than the greatest logical channel number (lcnhi) defined in the LCGDEF=(lcg,lcnhi) keyword on X25.MCH.

**Utility Action:** Verify the number of virtual circuits to be defined by this X25.VC or X25.LINE for the corresponding logical channel group (lcg). The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL055I: FROM THE SNA RESOURCE NUMBER aaa, THE COMBINATION OF bbb AND THE SNA RESOURCE NUMBER EXCEEDS 8 DIGITS, REQUIRED**

**Explanation:** You can specify a unique rule for statement name creation (called SNA name) using the following keywords: PRFLINE, PRFPU, PRFLU, and SUFFIX.

The name results from the concatenation of the PRFxxx value and the SUFFIX number, and must not exceed 8 characters. For example, if you code: X25.VC LCN=(1,10),...,PRFLINE=CIRCUIT,SUFFIX=1, the names CIRCUIT1 to CIRCUIT10 are generated. The name CIRCUIT10 induces an error, because it contains 9 characters.

**Utility Action:** Verify that the association of PRFxxx and SUFFIX matches with the maximum number of virtual circuits or resources defined. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL056I: LLC=LLCn INHERITED VALUE FROM aaa STATEMENT, INVALID WITH bbb STATEMENT, REQUIRED**

**Explanation:** The LLC type for PVC can be coded on X25.NET or X25.MCH statements and is sifted to any PVC if the LLC type is omitted on X25.LINE or X25.VC. Because LLC type 2 and 3 are not supported by X25.VC leased, LLC=LLC2 or LLC=LLC3 cannot be sifted to X25.VC leased.

**Utility Action:** Code the accurate LLC type on the X25.VC statement. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL057I: THE NUMBER OF X25.MCHS CODED IS LESS THAN THE NUMBER SPECIFIED ON aaa , CODED ON THE bbb STATEMENT**

**Explanation:** The number of X25.MCH statements coded is not correct:

1. Where aaa=X25.MCHCNT and bbb=BUILD, the number of physical circuits defined by X25.MCHCNT in an NPSI generation deck must match the number of X25.MCH statements coded.
2. Where aaa=SCOPE and bbb=X25.EXIGMCH, the number of NPSI DTEs coded in the XI resource definition must match the number of X25.MCH statements coded.

**Utility Action:**

1. Code one X25.MCH statement for each X25.MCHCNT physical circuit.
2. Code one X25.MCH statement for each DTE address defined in the X25.EXIGMCH SCOPE list.

The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL058I: BE AWARE THAT FROM NPSI V3R3 X25.IDNUMH SPECIFIES THE FIRST DIGIT OF THE IDNUM USED TO MAP NON-SNA RESOURCES, THEREFORE THE LEFTMOST DIGIT 0 IS IGNORED.**

**Explanation:** The X25.IDNUMH has been coded under the form: 01. The leftmost digit 0 is ignored to build the IDNUM for the switched major node PU which will be 1XXXXXX.

**Utility Action:** This is an informational message.

**Programmer Action:** Verify the IDNUM of the switched major node.

---

**BAL059I: aaa[(x)] = bbb INVALID, n CHARACTERS REQUIRED**

**Explanation:** The value bbb specified for keyword aaa is not valid, you must code n characters. If (x) is present, it shows the subkeyword position of bbb within keyword aaa.

**Utility Action:** Verify the number of characters to be coded, no default is assumed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL060I: aaa [(x)] = bbb INVALID, NOT WITHIN RANGE ccc, REQUIRED**

**Explanation:** The value bbb specified for keyword aaa is not within the range of values authorized. If (x) is present, it shows the subkeyword position of bbb within keyword aaa.

**Utility Action:** Verify the valid range, no default is assumed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL061I: aaa [(x)] = bbb INVALID, REQUIRED**

**Explanation:** The value bbb specified for keyword aaa is not valid. If (x) is present, it shows the subkeyword position of bbb within keyword aaa.

**Utility Action:** Verify the valid value allowed, no default is assumed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL062I: THE NUMBER OF aaa SUBVALS CODED MUST BE THE SAME AS bbb SUBVALS CODED,REQUIRED**

**Explanation:** The number of subkeywords coded on keyword aaa is not the same number of subkeywords coded on bbb keyword. For example: AEFCTCP/AEFBYTE, CTCP/CUD0...

**Utility Action:** Verify the value coded, no default is assumed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL063I: PARAMETERS CONFLICT, aaa = bbb ONLY ALLOWED IF ccc = ddd, REQUIRED**

**Explanation:** Operand aaa = bbb can only be coded if keyword ccc = ddd is specified.

**Utility Action:** Check if keyword bbb can be coded, this condition is required for validity of keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL064I: PARAMETERS CONFLICT, aaa ON bbb REQUIRED FOR AN XI MCH**

**Explanation:** Operand aaa is missing or is coded with an incorrect value, either this keyword or another value is required on a MCH dedicated to XI.

**Utility Action:** Replace the keyword aaa with the correct value. This condition is required for the validity of the XI MCH. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.



---

**BAL065I: PARAMETERS CONFLICT, FRMLGTH ON X25.MCH MUST MATCH  
FRMLGTH(x) PKTMODL, PKTMODL(x) ON X25.EXIGMCH, REQUIRED**

**Explanation:** This is an XI MCH; the FRMLGTH or PKTMODL value coded does not match with the value found in the same position corresponding to the address value found in the SCOPE (X25.EXIMMCH) sublist.

**Utility Action:** Replace the correct value for keyword aaa. This condition is required for the validity of the XI MCH. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL066I: Reserved.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

---

**BAL067I: PARAMETERS CONFLICT, LCN = aaa INVALID IF LCGN IS INVALID OR NOT  
DEFINED IN X25.MCH(LCGDEF), REQUIRED**

**Explanation:** Operand LCN (X25.LINE or X25.VC) is not valid; this logical channel number is coded under a logical channel group number (X25.LCGN LGN=xx), which is not defined in the logical channel group definition (X25.MCH ...LCGDEF=...).

**Utility Action:** Check the logical channel group used in the corresponding MCH. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL068I: OSITYPE=n INVALID, OSITYPE VALUE MUST BE LESS THAN OR EQUAL TO  
THE MAXIMUM CTCP NUMBER WITH WHICH THE X25.MCH CAN COMMUNI-  
CATE, OSITYPE=0 ASSUMED FOR STATEMENT KEYWORD VALIDATION.**

**Explanation:** The value n of the OSITYPE keyword must be less than or equal to the greatest value in the CTCP sublist coded on the X25.MCH statement.

**Utility Action:** Verify the value coded. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL069I: aaa [(x)] = bbb INVALID, DUPLICATES NOT ALLOWED, REQUIRED**

**Explanation:** The value bbb specified for keyword aaa is not valid, this value has already been coded inside the sublist. Duplicate values inside the sublist are not allowed. The (x) shows the subkeyword position of bbb within the keyword aaa.

**Utility Action:** Verify the value coded, no default is assumed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL070I: Reserved.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

---

**BAL071I: SEQUENCE ERROR, aaa INVALID WHEN bbb SPECIFIED ON ccc, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The statement aaa cannot be coded when keyword bbb is coded on statement ccc.

Example:

```
X25.LU is not allowed on switched
virtual circuit X25.LINE...TYPE=S)
```

**Utility Action:** The statement is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL072I: PARAMETERS CONFLICT, PKTMODL = n INVALID WHEN FRMLGTH = m DUE TO PACKET HEADER LENGTH, REQUIRED**

**Explanation:** NPSI supports packet sizes from 32 to 4096 bytes. The packet modulo value n does not match with the maximum frame length m. The maximum frame length, in bytes, that flows over the physical circuit is the data packet maximum length added to the packet header length. Packet header length is:

```
3 bytes for packet modulo 8
4 bytes for packet modulo 128
```

Therefore:

```
for PKTMODL=8 valid FRMLGTH values are 35 to 4099
for PKTMODL=128 valid FRMLGTH values are 36 to 4100.
```

**Utility Action:** PKTMODL value must match with the FRMLGTH value, which is required for statement validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL073I: LLC KEYWORD OMITTED, LLC=LLCn INHERITED VALUE FROM aaa**

**Explanation:** The LLC keyword is mandatory on a permanent virtual circuit. It can be coded on X25.NET or X25.MCH and sifted to X25.LINE or X25.VC. You must verify the compatibility of the LLC type sifted with the support function defined in the MCH (PAD,GATE...). This error message is followed by another one indicating the mismatch between the LLC type sifted and the support function chosen (BAL048I,BAL051I).

**Utility Action:** The LLC keyword must be coded on the current permanent virtual circuit. The generation process ends after validation of the generation definition. This is an informational message. The generation process is not interrupted.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL074I: Reserved.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

---

**BAL075I: LCN(x)= n INVALID, THE LCN CODED IS GREATER THAN THE HIGHER VALUE CODED IN X25.MCH(LCGDEF), REQUIRED**

**Explanation:** The LCN value n coded is greater than the greatest logical channel number (lcnhi) defined in the LCGDEF=(lcn,lcghi) keyword on X25.MCH (x), which shows the subkeyword position of the LCN value.

**Utility Action:** Verify the number of virtual circuits to be defined by this X25.VC for the corresponding logical channel group (lcn). The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL076I: PARAMETERS CONFLICT, aaa = n INVALID WHEN LCN0=USED IS CODED ON X25.MCH. THE FIRST aaa MUST BE EQUAL TO 0, REQUIRED**

**Explanation:** You have specified LCN0 = USED (or defaulted) in the above X25.MCH. The aaa keyword (LCGN in X25.LCG or LCN in X25.LINE/X25.VC) is not valid.

- The first X25.LCG must define the logical channel group number 0.
- The first virtual circuit in that LCGN 0 must define the logical channel number 0.

**Utility Action:** Verify if your PSDN does or does not use the value 0 on logical group 0. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL077I: aaa INVALID, n SUBOPERAND(S) REQUIRED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The number of subvalues coded for keyword aaa is incorrect. Exactly n subvalues must be coded.

**Utility Action:** Operand aaa is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL078I: SEQUENCE ERROR, aaa MUST BE PROCESSED AFTER bbb IN AN X25 NCP PACKET SWITCHING INTERFACE (NPSI) GENERATION, REQUIRED**

**Explanation:** The statement aaa is misplaced among the generation input deck; it must follow a bbb statement.

**Utility Action:** The statement aaa is not processed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL079I: aaa INVALID, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The value coded for keyword aaa is not a valid specification. For keyword PADPARM, the value coded must be a string of hexadecimal digits.

**Utility Action:** Operand aaa is ignored for statement keyword validation. The generation process ends after validating the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL080I: IN THE PREVIOUS MCH, THE NUMBER OF SVC DEDICATED TO SAD FUNCTION MUST BE EQUAL TO THE VALUE CODED IN SVCINN(X25.MCH), REQUIRED**

**Explanation:** An MCH can be associated with the switched virtual circuit subarea communication (SVCSC) or SAD function. This function requires two pairs of virtual circuit definitions:

- The SNA set of resources defined with X25.LINE and X25.PU statement pairs coded with TYPE=S, DSTNODE=INN, PUTYPE=4, no LCN coded.
- The X25 set of resources defined with X25.LINE and X25.PU statement pairs or X25.VC statements coded with LCN=xm, defining the virtual circuits to NPSI.

The following problem occurs in the previous MCH. SVCINN (which represents the number of switched virtual circuits allocated to subarea node traffic) must equal the number of SNA resources. DSTNODE=INN, PUTYPE=4.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL081I: IN THE PREVIOUS MCH, SVCINN(X25.MCH) VALUE MUST BE LESS THAN OR EQUAL TO THE NUMBER OF SWITCHED VIRTUAL CIRCUITS IN THIS MCH, REQUIRED**

**Explanation:** An MCH can be associated with the SVCSC (or SAD) function. This function requires two pairs of virtual circuit definitions:

- The SNA set of resources defined with X25.LINE and X25.PU statement pairs coded with TYPE=S, DSTNODE=INN, PUTYPE=4, and no LCN coded.
- The X25 set of resources defined with X25.LINE and X25.PU statement pairs or X25.VC statements coded with LCN=x defining the virtual circuits to NPSI.

The following problem occurs in the previous MCH. SVCINN (which represents the number of switched virtual circuits allocated to subarea node traffic) must be less or equal to the number of X25 resources.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL082I: IN THE PREVIOUS MCH, IF SVCINN(X25.MCH) VALUE IS EQUAL TO THE NUMBER OF SWITCHED VIRTUAL CIRCUITS IN THIS MCH, THEN NO OTHER LLC TYPE THAN LLC3 CAN BE SELECTED, REQUIRED.**

**Explanation:** An MCH can be associated with the SVCSC (or SAD) function. This function requires two pairs of virtual circuit definitions:

- The SNA set of resources defined with X25.LINE and X25.PU statement pairs coded with TYPE=S, DSTNODE=INN, PUTYPE=4, no LCN coded.
- The X25 set of resources defined with X25.LINE and X25.PU statement pairs or X25.VC statements coded with LCN=x defining the virtual circuits to NPSI.

The following problem occurs in the previous MCH. SVCINN (which represents the number of switched virtual circuits allocated to subarea node traffic) pairs with the number of X25 resources. As SVCSC is using LLC type 3, the LLCLIST must contain only LLC3.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL083I: IN THE PREVIOUS MCH, SVCINN(X25.MCH) SPECIFIED NOT NULL BUT NO SVC DEDICATED TO THE SAD FUNCTION IN THIS MCH, REQUIRED**

**Explanation:** An MCH can be associated with the SVCSC (or SAD) function. This function requires two pairs of virtual circuit definitions:

- The SNA set of resources defined with X25.LINE and X25.PU statement pairs coded with TYPE=S, DSTNODE=INN, PUTYPE=4, no LCN coded.
- The X25 set of resources defined with X25.LINE and X25.PU statement pairs or X25.VC statements coded with LCN=x defining the virtual circuits to NPSI.

The following problem occurs in the previous MCH. SVCINN (which represents the number of switched virtual circuits allocated to subarea node traffic) defines n SNA resources that are not coded in this MCH.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL084I: LLCLIST MUST CONTAIN ONLY LLC4 WHEN aaa IN THIS MCH, REQUIRED**

**Explanation:** The LLCLIST keyword (X25.MCH) must contain only LLC type 4 when the MCH is dedicated to the aaa function (example: fast connect option)

**Utility Action:** The LLCLIST keyword is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL085I: aaa = bbb INVALID, NOT WITHIN RANGE ccc, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The value bbb specified for keyword aaa is not within the valid range of values ccc.

**Utility Action:** The aaa keyword is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL086I: aaa = bbb INVALID, THE bbb VALUE MUST NOT BE GREATER THAN THE NUMBER OF SVC BELONGING TO THIS MCH, REQUIRED**

**Explanation:** The value bbb specified for keyword aaa must be less than or equal to the number of switched virtual circuits defined for the physical circuit.

Example: QTY keyword on X25.FCG.

**Utility Action:** The aaa keyword is required, no default value is assumed. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL087I: PARAMETERS CONFLICT, THIS MCH IS FULLY DEDICATED TO aaa FUNCTION, bbb STATEMENT IS NOT ALLOWED**

**Explanation:** This MCH supports the aaa function, then the bbb statement cannot be coded.

Example: X25.LINE statement cannot be coded when the MCH supports the fast connect function (CUD0 or SUBD).

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL088I: PARAMETERS CONFLICT, NO aaa STATEMENT CAN BE DEFINED UNDER THIS bbb STATEMENT WHEN ccc = ddd**

**Explanation:** The statement aaa cannot be coded after the statement bbb where the keyword ccc is equal to ddd.

Example: X25.LINE statement cannot be coded after X25.MCH where CONNECT=CUD0.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL089I: aaa OMITTED IN bbb, REQUIRED WHEN ccc SPECIFIED**

**Explanation:** Operand aaa is not specified in keyword bbb. It is required if keyword bbb is coded.

Example: When the subaddressing is used to select the LLC type 4, LLC4 must be coded in the keyword LLCLIST.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL090I: IN THE PREVIOUS MCH, THE CTCP NUMBER n HAS NO X25.FCG STATEMENT CODED, REQUIRED**

**Explanation:** The previous MCH is dedicated to the fast connect function. The CTCP number n has no corresponding X25.FCG; if more than one destination CTCP is attached to a particular MCH, an X25.FCG statement must be coded for each CTCP.

**Utility Action:** Define an X25.FCG for each CTCP of the MCH. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL091I: aaa = bbb MUST BE UNIQUE AMONG THE ccc STATEMENTS, REQUIRED**

**Explanation:** The bbb value of keyword aaa has already been defined in a similar ccc statement. Example: A CTCP number must be unique among the X25.FCG or X25.CPL statements.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL092I: aaa = bbb INVALID, THE bbb VALUE IS GREATER THAN THE HIGHEST VALUE CODED IN ccc, REQUIRED**

**Explanation:** The bbb value of keyword aaa is outside the range defined by the greatest value of keyword ccc. Example: The CTCPNO value coded in X25.FCG cannot be greater than the greatest CTCP number code in the X25.MCH.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL093I: IN THE PREVIOUS MCH, ONLY n SNA SESSIONS (SUM OF QTY VALUES) DEFINED, IT SHOULD BE AT LEAST EQUAL TO THE m SVCS DEFINED, REQUIRED**

**Explanation:** There are fewer SNA sessions than virtual circuits in the previous MCH allocated to fast connect. The SNA resources are defined by the sum of the QTY values of the X25.FCG coded under this MCH. The X25 resources are equal to the number of virtual circuits defined in the X25.VC statement. Because of the way connections are established, the number of SNA resources must be at least equal to the number of X25.resources.

**Utility Action:** Verify the number of resources defined. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the definition of resources and resubmit the job.

---

**BAL094I: CTCP [(x)] = n INVALID WITH CONNECT OR FTPI= CUD0, REQUIRED SUBD**

**Explanation:** The CTCP value coded is not valid; the range 80–85 is reserved for the LLC type selection and is exclusive to the fast connect or FTPI function. If (x) is present, it shows the subkeyword position within the CTCP keyword.

**Utility Action:** Remove the incorrect value. The generation process ends after validation of the generation definition.

**Programmer Action:** Supply the missing keyword and resubmit the job.

---

**BAL095I: NPSI FEATURE LEVEL (n) DOES NOT CORRESPOND TO THE NPSI FEATURE LEVEL ORDERED, OR THE NPSI MACLIB IS MISSING IN THE DATA-SETS LIBRARY, REQUIRED**

**Explanation:** Either the X25.USGTIER value coded exceeds the installed feature level, or the NPSI MACLIB is missing in the data sets library. The X25.USGTIER value must be lower than or equal to the feature level ordered.

**Utility Action:** Check either the feature level installed or the data sets library.

**Programmer Action:** Resubmit the job with the correct level.

---

**BAL096I: NPSI FEATURE LEVEL (X25.USGTIER=n) IS NOT COMPATIBLE WITH THE NCP FEATURE LEVEL ORDERED (USGTIER= m), REQUIRED**

**Explanation:** The X25.USGTIER value coded is less than the usage tier value specified by the USGTIER keyword of the NCP BUILD statement. The X25.USGTIER value must be greater than or equal to the NCP USGTIER value.

**Utility Action:** Check the feature level installed.

**Programmer Action:** Resubmit the job with the correct level.

---

**BAL097I: PARAMETERS CONFLICT, NO aaa KEYWORD CAN BE CODED WHEN THIS bbb IS ALLOCATED TO ccc FUNCTION**

**Explanation:** The keyword aaa coded on the bbb statement is incompatible with the function ccc. Example: The LLC keyword is not allowed on X25.LINE specifying a SVCSC resource.

**Utility Action:** Remove the keyword aaa. The generation process ends after validation of the generation definition.

**Programmer Action:** Remove the keyword aaa and resubmit the job.

---

**BAL098I: aaa [(x)] = bbb INVALID, IS NOT AN HEXADECIMAL VALUE, REQUIRED**

**Explanation:** The value bbb specified for keyword aaa is not valid; it must be coded in hexadecimal. If (x) is present, it shows the subkeyword position of bbb within the keyword aaa.

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL099I: NSTDFAC [(x)] = nn INVALID, IS A DUPLICATE OF A STANDARD FACILITY, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The nonstandard facility nn is not valid; it has already been coded in the sublist. The (x) shows the subkeyword position of the duplicate facility.

**Utility Action:** The NSTDFAC keyword is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL100I: Not used.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

---

**BAL101I: Not used.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

---

**BAL102I: COMBINATION OF aaa AND bbb EXCEEDS 8 DIGITS, REQUIRED**

**Explanation:** There is an 8-character limitation on labels; the resource name created by appending the SUFFIX value bbb to the appropriate PRFLINE, PRFPU, or PRFLU value aaa exceeds 8 characters.

**Utility Action:** The resource name is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.



---

**BAL103I: FROM THE SNA RESOURCE NUMBER n, THE COMBINATION OF aaa AND THE SNA RESOURCE NUMBER EXCEEDS THE NUMBER OF DIGITS CODED (PREFIX + SUFFIX), REQUIRED**

**Explanation:** The length of the resource name created by appending the SUFFIX to the appropriate PRFLINE, PRFPU, or PRFLU value aaa must be constant. In the case of X25.VC or X25.FCG the SUFFIX keyword defines the lower limit of the resource number; this number is increased depending of the LCN or QTY keyword , then the SUFFIX length can be changed.

**Utility Action:** The resource name is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL104I: aaa NOT ALLOWED WITH bbb FUNCTION, FOLLOWING aaa's IGNORED**

**Explanation:** Keyword aaa is not allowed on virtual circuits supporting the bbb function. Example: IDNUMT keyword cannot be coded on virtual circuits of an MCH dedicated to fast connect or FTPI.

**Utility Action:** Keyword aaa is ignored on the following virtual circuits. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL105I: IDNUMT=aaa INVALID, DUPLICATE IDBLK/IDNUMT PAIRS ARE NOT ALLOWED, IGNORED**

**Explanation:** The aaa value for keyword IDNUMT is not valid, it has already been coded. The result leads to a duplicate SNA resource name (combination of IDBLK(C-G-P) keyword value and IDNUMT keyword value).

**Utility Action:** Keyword IDNUMT is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL106I: IDNUMT = aaa INVALID, IDNUMT MUST BE EVEN, REQUIRED**

**Explanation:** The aaa value for keyword IDNUMT is not valid; IDNUMT must be an even number.

**Utility Action:** Keyword IDNUMT is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL107I: ON THE PREVIOUS MCH, NO SVC WAS DEFINED WHILE IDBLK HAD BEEN CODED**

**Explanation:** IDBLK was coded to define switched resources, but no switched virtual circuit was coded under this MCH.

**Utility Action:** Verify the generation definition. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL108I: PARAMETER CONFLICT, WHEN ID IS USED, LLC0 WITH SUBADDR=NO AND WITHOUT GATE=DEDICAT, AND LLC2 OR LLC3 WITH SVCINN=0, CANNOT BE SPECIFIED AT THE SAME TIME ON THIS MCH, REQUIRED**

**Explanation:** The NCP anonymous caller function is supported through the use of ID keyword on X25.VC and X25.LINE. This applies to MCHs supporting only one kind of SVC:

- Peripheral SNA node LLC type 2 or LLC type 3
- or
- PCNE (LLC type 0) SVCs without DATE or subaddressing.

**Utility Action:** The LLC type are exclusive in the LLCLIST. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL109I: THIS MCH IS DEDICATED TO INTERWORKING CONNECTIONS FOR DIAL OUT BY THE PSDN, ONLY 1 SVC IS ALLOWED, REQUIRED.**

**Explanation:** More than one SVC is coded. However, only one SVC is allowed for the interworking connection for dial-out-by-the-PSDN.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL110I: aaa NOT CODED, CONFLICTING OPTION, REQUIRED**

**Explanation:** The aaa keyword is missing on the current statement. The MCH is dedicated to a specific function, then the keyword aaa must be coded on all the virtual circuits defined. (Example: support of NCP anonymous caller function the ID keyword must be coded on all SVCs for the given MCH).

**Utility Action:** The keyword aaa is required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL111I: aaa = bbb INVALID, n DIGITS ARE REQUIRED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The number of digits in the keyword value bbb is not valid; n digits must be coded.

**Utility Action:** The keyword aaa is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL112I: USER(parm1) TABLE INCOMPLETELY DEFINED, 32 KEYWORDS ARE NEEDED ON AN X25.TRAN STATEMENT, REQUIRED**

**Explanation:** USER(x) specifies the user translate tables. Two tables can be specified; sixteen keywords are required to complete the definition of a user table. Table definition x is incomplete, one or several keywords are missing.

**Utility Action:** Complete the table user x to validate the X25.TRAN statement. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL113I: PARAMETERS CONFLICT, aaa = bbb INVALID WHEN ccc = ddd AND eee = fff SPECIFIED, IGNORED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** The value bbb is not valid for keyword aaa when keyword ccc = ddd and keyword eee = fff.

**Utility Action:** The keyword aaa is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL114I: aaa = bbb OMITTED, WHEN ccc = ddd AND eee = fff SPECIFIED, REQUIRED.**

**Explanation:** The keyword aaa must be coded with the value bbb when keyword ccc = ddd and keyword eee = fff.

**Utility Action:** The keyword aaa = bbb is required for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL115I: IDNUMT = xx IS NOT COMPATIBLE WITH THE LCN VALUE, IGNORED ID**

**Explanation:** ID and IDNUMT keywords. When coded on X25, the generation of IDNUMT or ID values is automatically done by NPSI from the IDNUMT or ID value. This value is incremented for each virtual circuit generated (on the LCN basis). On the current X25.VC, the greatest value of range for ID (FFFFFFF) or IDNUMT (FFFF) has been reached; the LCN keyword defines too many virtual circuits for the starting value of ID or IDNUMT.

**Utility Action:** Check the ID or IDNUMT starting value for the LCN definition. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL116I: IN THE PREVIOUS MCH, LSPRI=YES(X25.MCH) WAS CODED BUT NO PVC AND NO SVC IS ABLE TO SUPPORT THE LSPRI FUNCTION, REQUIRED**

**Explanation:** The link session priority is required for peripheral lines (boundary virtual circuits working under LLC type 2 or LLC type 3), but neither permanent virtual circuits does support LLC type 2 or LLC type 3 nor switched virtual circuits LLC type 2 or LLC type 3 in LLCLIST.

**Utility Action:** At least one virtual circuit must be defined with this LLC type. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL117I: IN THE PREVIOUS MCH, THE LLCLIST KEYWORD MUST CONTAIN ALL THE LLC TYPE SPECIFIED FOR SUBADDRESSING FUNCTION, REQUIRED**

**Explanation:** A selected LLC type by the subaddressing function has been omitted in the LLCLIST.

**Utility Action:** All the LLC types selected by subaddressing function must be coded in the LLCLIST keyword. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL118I: IN THE PREVIOUS MCH, THE NUMBER OF SVCS DEDICATED TO THE SAD FUNCTION IS EQUAL TO THE NUMBER OF SVCS IN THIS MCH, THEN THE SUBADDRESSING FUNCTION CAN ONLY APPLY ON LLC3**

**Explanation:** All the switched virtual circuits belong to the SVCSC function; SVCSC connection is supported only by LLC type 3; therefore, only LLC3 can be defined in LLCLIST and consequently the LLC type selection by subaddressing can only apply to LLC type 3.

**Utility Action:** Remove all LLC types except LLC3 from LLCList. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL119I: PAD SUPPORT IS DEFINED ON THE PREVIOUS MCH BUT NEITHER PVC WITH LLC=LLC5 CODED NOR SVC TYPE 5 (LLCLIST=LLC5) CODED.**

**Explanation:** You have coded PAD=INTEG or PAD=TRANSP on X25.MCH and no virtual circuit is eligible for LLC type 5.

**Utility Action:** This is an informational message.

**Programmer Action:** Verify your generation definition.

---

**BAL120I: aaa = bbb INVALID, NOT DEFINED ON ccc, REQUIRED**

**Explanation:** Value bbb for keyword aaa is not valid. It must be in relation and coded in a previous statement.

Example: CTCP number n defined in a FTPI MCH must be defined in a corresponding X25.CPL...CTCPNO=n.

**Utility Action:** Operand aaa = bbb is required for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL121I: IDNUMT NOT ALLOWED, NEITHER ASSOCIATION OF aaa OR OF bbb ID CODED IN X25.MCH, NOR ccc CODED IN LLCLIST, REQUIRED**

**Explanation:** Conditions to support the IDNUMT or ID assignment are not filled.

For IDNUMT either:

- LLC0 in LLCLIST and GATE=DEDICAT
- LLC0 in LLCLIST and GATE=GENERAL and SUBADDR=NO
- LLC4 in LLCLIST
- LLC5 in LLCLIST

For ID either:

- LLC0 in LLCLIST and GATE=NO
- LLC0 in LLCLIST and GATE=GENERAL and SUBADDR=NO
- LLC2 in LLCLIST
- LLC3 in LLCLIST and SVCINN=0

**Utility Action:** Fill in the conditions required. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

#### **BAL122I: aaa NOT ALLOWED WITH bbb FUNCTION**

**Explanation:** The aaa keyword is not allowed when the involved MCH or virtual circuit is dedicated to a specific function bbb. For example, to support the NCP anonymous aaller function, the ID keyword cannot be coded on SVCs for a given MCH that supports either the FTPI or fast connect function.

**Utility Action:** The keyword aaa is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

#### **BAL123I: IDNUMT OR ID KEYWORD IN CONFLICT WITH OPTION CHOSEN. IDNUMT ASSIGNMENT OPTION BY THE IDNUMT KEYWORD APPLIES ON THE WHOLE GENERATION. ID ASSIGNMENT OPTION BY THE ID KEYWORD APPLIES ON ALL SVCs OF A MCH.**

**Explanation:**

- IDNUMT keyword is missing on this non-SNA SVC. The IDNUM assignment for non-SNA SVCs by the IDNUMT keyword is a user option that must apply on all the non-SNA SVCs of the NPSI generation.
- ID keyword is missing on this SVC. The IDNUM assignment for SNA SVCs or PCNE SVCs, without subaddressing or DATE by the ID keyword, applies to all SVCs on a given MCH.

**Utility Action:** Add the missing keyword. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

#### **BAL124I: Not used.**

**Explanation:** Not applicable.

**Utility Action:** Not applicable.

**Programmer Action:** Not applicable.

#### **BAL125I: LOGAPPL NOT CODED, FOR CONTROL SESSION ESTABLISHMENT THE ACTIVATION REQUEST IS SUPPOSED TO BE DONE BY THE APPLICATION PROGRAM(S)**

**Explanation:** LOGAPPL is an optional keyword, not coded on this MCH. The control session for GATE can be established by the network operator or by the application program. If this is done by the network operator, the LU simulator must have knowledge of the LOGAPPL name, then LOGAPPL must be coded.

**Utility Action:** This is an informational message to advise you to verify the virtual circuit establishment method.

**Programmer Action:** See previous message.

---

**BAL126I: LOGAPPL[(x)]NOT CODED, FOR CONTROL SESSION ESTABLISHMENT THE ACTIVATION REQUEST IS SUPPOSED TO BE DONE BY THE APPLICATION PROGRAM**

**Explanation:** LOGAPPL is an optional keyword, not coded for the position x inside the sublist. The control session for gate can be established by the network operator or by the application program. If this is done by the network operator, the LU simulator must have knowledge of the LOGAPPL name, then LOGAPPL must be coded.

**Utility Action:** This is an informational message to advise the user to verify the virtual circuit establishment method.

**Programmer Action:** See previous message.

---

**BAL127I: aaa = bbb INVALID, WHEN ccc NOT CODED OR INVALID ON ddd, ccc ASSUMED FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** Value bbb for keyword aaa is not valid. It is valid only if ccc is coded on the ddd statement. You can code only aaa = ccc.

**Utility Action:** Operand aaa = eee is required for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL128I: IN THE PREVIOUS MCH (ADDR=aa) NPPVCN=n INVALID, IT MUST NOT EXCEED THE TOTAL NUMBER OF VIRTUAL CIRCUITS CODED UNDER THE MCH (m FOR THE MCH INVOLVED)**

**Explanation:** NPPVCN defines the maximum number of virtual circuits on which you need performance data reporting at the same time. The value coded n is not valid: it exceeds the total number m of virtual circuits coded under that MCH.

**Utility Action:** Check the value coded related to the number of virtual circuits. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL129I: WHEN NPM ACCOUNTING FUNCTION IS CALLED ON BUILD STATEMENT (X25.BYTTHROD AND X25.SEGTHROD), AT LEAST ONE MCH MUST REPORT ACCOUNTING DATA (ACCOUNT KEYWORD REQUIRED)**

**Explanation:** NPM for NPSI is triggered by X25.BYTTHROD or X25.SEGTHROD keywords. The keyword ACCOUNT=YES or a selection of LLC types must be coded on at least one MCH.

**Utility Action:** Verify your NPM requirement. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL130I: aaa VALUE CODED ON X25.PU NO LONGER RECOGNIZED BY NPSI. THIS PU IS ELIGIBLE FOR DATA COLLECTION BY NPM WHEN NPACOLL=VCPU HAS BEEN CODED ON X25.MCH. aaa VALUE IS REPLACED BY bbb.**

**Explanation:** The keyword aaa is not available; it has been replaced by the value bbb coded or defaulted on the ccc statement.

**Utility Action:** This is an informational message.

**Programmer Action:** None.

---

**BAL131I: aaa ROUTINE NAME CAN NOT BE SPECIFIED ON bbb KEYWORD WHEN ALL THE X25.MCH CODED IN THE GENERATION HAVE parm3**

**Explanation:** The aaa routine cannot be called by the bbb keyword on GENEND statement. The condition ccc is not fulfilled in at least one MCH. For example, in the statement GENEND ...UACCTNG=BALANPM,..., the routine BALANPM is called for NPM accounting; if ACCOUNT=NO is coded or defaulted in all the MCHs, this routine cannot be called.

**Utility Action:** The keyword aaa is ignored for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL132I: IN THE PREVIOUS MCH (ADDR= aaa), LCGN= n IS DEFINED ON X25.MCH(LCGDEF), THE CORRESPONDING X25.LCGN STATEMENT IS MISSING, REQUIRED**

**Explanation:** The logical channel group number used by the controller is defined in the LCGDEF keyword (X25.MCH). Each LCGN defined must correspond to a X25.LCGN statement.

**Utility Action:** The generation process ends after validation.

**Programmer Action:** Add the missing X25.LCGN statement keyword validation. Correct the error and resubmit the job.

---

**BAL133I: IN THE PREVIOUS MCH (ADDR= aaa), THE HIGHEST LOGICAL CHANNEL NUMBER DEFINED IN LCGDEF FOR LCGN=n IS m, THEN A CORRESPONDING X25.LINE OR X25.VC MUST BE CODED FOR THE PRECEDING LCGN, REQUIRED**

**Explanation:** The greatest logical channel number for LCGN n has been defined to m in the LCGDEF keyword of X25.MCH. The corresponding Logical Channel number must be defined by an X25.LINE or X25.VC, depending on which one is missing.

**Utility Action:** The generation process ends after validation.

**Programmer Action:** Add the missing X25.LINE or X25.VC statement. Correct the error and resubmit the job.

---

**BAL134I: IN THE PREVIOUS MCH (ADDR= aaa), ACCOUNTING REQUEST INVALID FOR LLCx (ACCOUNT=LLCx), THIS LLC TYPE IS NEITHER SUPPORTED IN THE LLCLIST FOR SVCS NOR SUPPORTED BY ANY PVCS(LLC KEYWORD)**

**Explanation:** NPM accounting is requested on virtual circuits of LLC type x ; no virtual circuit of this type is defined under the previous MCH aaa involved (neither PVC nor SVC).

**Utility Action:** Verify your NPM request.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL135I: IN THE PREVIOUS MCH (ADDR=aaa), ACCOUNTING REQUEST INVALID FOR LLCx (ACCOUNT=LLCx), THIS LLC TYPE IS NOT SUPPORTED IN THE LLCLIST FOR SVCS**

**Explanation:** NPM accounting is requested on virtual circuits of LLC type x; no virtual circuit of this type is defined under the previous MCH aaa involved. (Only SVCs are coded under this MCH, then the LLC type wanted by NPM must be included in the LLCLIST.)

**Utility Action:** Verify your NPM request.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL136I: IN THE PREVIOUS MCH (ADDR= aaa), ACCOUNTING REQUEST INVALID FOR LLCx (ACCOUNT=LLCx), THIS LLC TYPE IS NOT SUPPORTED BY ANY PVCS (LLC KEYWORD).**

**Explanation:** NPM accounting is requested on virtual circuits of LLC type x; no virtual circuit of this type is defined under the previous MCH aaa involved. (Only PVCs are coded under this MCH, then the LLC type wanted by NPM must be supported by at least one PVC.)

**Utility Action:** Verify your NPM request.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL137I: IN THE PREVIOUS MCH (ADDR=parm1), ACCOUNTING REQUEST INVALID FOR parm2, NO VC OF LLC TYPE 3 IS SUPPORTING parm3 DESTINATION NODE**

**Explanation:** NPM accounting is requested on virtual circuits of LLC type 3 BNN or INN. No virtual circuit of this type is defined under the previous MCH aaa involved.

**Utility Action:** Verify your NPM request.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL138I: aaa = bbb INVALID WHEN ccc NOT CODED ON ddd, REQUIRED**

**Explanation:** Value bbb for keyword aaa is not valid. It is valid only if ccc is coded on the ddd statement or keyword.

**Utility Action:** The condition is required for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL139I: aaa=bbb INVALID, ONLY ccc SUPPORTING ddd ALLOWED UNDER A MCH DEDICATED TO eee , REQUIRED**

**Explanation:** Value bbb for keyword aaa is not valid. The MCH is dedicated to the eee function, therefore only ccc supporting ddd is allowed. Example: LLC=LLC3 on X25.LINE is not valid when the MCH is dedicated to FTPI, only PVCs of LLC type 4 are allowed.

**Utility Action:** The condition is required for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL140I: PARAMETERS CONFLICT,WHEN THE NCP SLOWDOWN SPECIFIES 50 PERCENT OF THE BUFFER SIZE, THE UNSAFE LIMIT (INSLOW=(100,...) DEFINED BY NPSI CAN NOT BE 100 PERCENT HIGHER THAN THE NCP SLOWDOWN VALUE**

**Explanation:** INSLOW (X25.VCCPT) values are specified in relation to the values in the SLOWDOWN keyword of the NCP BUILD statement. The UNSAFE percentage (100%) of the NCP SLOWDOWN value (50) added to that SLOWDOWN value means that there is no BUFFER area free.

**Utility Action:** Reduce the INSLOW value. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.



---

**BAL141I: TIMER TICK SERVICE ROUTINE NO LONGER USED (EXCEPT WHEN FTPI FUNCTION), BALTICK ENTRY-NAME ROUTINE SUPPRESSED FROM TMRICK KEYWORD, IGNORED**

**Explanation:** For NPSI V3R4 and later releases, the TIMER TICK service routine is not used, except for the FTPI function.

**Utility Action:** This is an informational message. The generation process continues the validation of the generation definition.

**Programmer Action:** None.

---

**BAL142I: WHEN CAUSE AND DIAGNOSTIC BYTES IN RESET REQUEST PACKET ARE CHOSEN BY THE USER(CAUSE=DCE), THIS MCH MUST SUPPORT ONLY VIRTUAL CIRCUITS ON LEASED MODE AND LLC4 TYPE, REQUIRED**

**Explanation:** CAUSE=DCE coded on this MCH determines the type of cause and diagnostic byte that NPSI sends to the network in Reset and Restart Request packets. The type DCE is supported only for PVCs of LLC type 4.

**Utility Action:** Define virtual circuits according your request. The generation process continues the validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL143I: ALTCTCP (x)=n INVALID, CTCP=n ALREADY DEFINED AS PRIMARY CTCP IN THIS X25.CPL**

**Explanation:** An alternative CTCP number has been coded with the same number as the X25.CPL CTCP number (CTCPNO). The x indicates the position of the CTCP number among the sublist.

**Utility Action:** Suppress the wrong ALTCTCP number. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL144I: ALTCTCP=n HAS NOT BEEN DEFINED AS PRIMARY CTCP IN A PREVIOUS X25.CPL, REQUIRED**

**Explanation:** The CTCP number n has been coded as an alternative (ALTCTCPNO), but it has not been defined as CTCPNO in a X25.CPL.

**Utility Action:** ALTCTCP = n must be defined by a corresponding X25.CPL. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL145I: CTCPNO INVALID, THE FIRST X25.CPL STATEMENT MUST DEFINE THE CTCP NUMBER 0, REQUIRED**

**Explanation:** The first X25.CPL defines a CTCP number other than CTCP 0. The CTCP 0 must be the first CTCP defined by the X25.CPL.

**Utility Action:** X25.CPL ...CTCP=0... must be defined first. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL146I: IN THE PREVIOUS MCH (ADDR= aaa) BE AWARE THAT NPPVCH=n IS LOWER THAN THE TOTAL NUMBER OF VIRTUAL CIRCUITS CODED UNDER THE MCH (parm3 FOR THE MCH INVOLVED)**

**Explanation:** You want to perform NPM data collection on n virtual circuits at the same time. The n is less than the total number of virtual circuits in this MCH. If more than n virtual circuits are activated at the same time, you get a VTAM EXCEPTION 0812.

**Utility Action:** This is an informational message; do not activate at the same time more than n virtual circuits. The generation process continues the validation of the generation definition.

**Programmer Action:** None.

---

**BAL147I: SPEED, NOT USER CODED, IS DEFAULTED TO 9600 AND NPM IS USED ON THIS LINE; VERIFY THAT THIS VALUE IS ACCURATE TO ENSURE CORRECT CALCULATION OF LINE UTILIZATION.**

**Explanation:** The SPEED value is necessary for NPM; this value is defaulted to 9600. This could not be the right speed of the line and can induce a wrong calculation of line utilization.

**Utility Action:** This is an informational message. The generation process continues the validation of the generation definition.

**Programmer Action:** Verify that this value is accurate for NPM.

---

**BAL148I: aaa=bbb INVALID, INVALID DIGIT IN NUMBER, IGNORED FOR STATEMENT KEYWORD VALIDATION.**

**Explanation:** Value bbb is not a valid specification for keyword aaa. A character that is not valid has been coded in the decimal number bbb.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL149I: PRFLU KEYWORD HAS NO SENSE ON THIS SWITCHED VIRTUAL CIRCUIT SINCE NO LU STATEMENT IS GENERATED FOR SWITCHED VIRTUAL CIRCUIT.**

**Explanation:** PRFLU coded on this X25.VC defining a switched virtual circuit has no sense. No LU is generated for switched circuit, then no name is generated.

**Utility Action:** This is an informational message. The generation process continues the validation of the generation definition.

**Programmer Action:** None.

---

**BAL150I: X25.MAXPIU VALUE SHOULD NOT EXCEED THE PRODUCT OF NCP BUFFER SIZE(BFRS KEYWORD ON BUILD) BY 255, X25.MAXPIU = n IS REPLACED BY THAT NCP BUFFERS CAPACITY: MAXPIU = m.**

**Explanation:** X25.MAXPIU value is greater than the NCP buffer's capacity. The X25.MAXPIU value n is replaced by the product of the NCP buffers size by 255.

**Utility Action:** This is an informational message. The generation process continues the validation of the generation definition.

**Programmer Action:** None.

---

**BAL151I: MONLINK = aaa WHILE XMONLNK=NO ON X25.MCH; BE AWARE THAT MONLINK=YES, OR CONT, OR CONTINUOUS CAN REQUIRE XMONLNK=YES TO BE CODED ON THE ASSOCIATED X25.MCH FOR SPECIFIC FUNCTIONS (REMOTE LOADING ACTIVATION...).**

**Explanation:** MONLINK is coded YES or CONT or CONTINUOUS while XMONLNK=NO on X25.MCH. Some specific functions like RLA can require XMONLNK=YES.

**Utility Action:** This is an informational message. The generation process continues the validation of the generation definition.

**Programmer Action:** See Chapter 4, "Defining X.25 Networks."

---

**BAL152I: aaa = bbb NOT ALLOWED ON ccc WHEN THIS ddd IS ALLOCATED TO eee**

**Explanation:** The keyword aaa = bbb is not allowed on a statement defining a specific function. For example, FTPI=YES is not allowed on an X25.MCH allocated to XI—NPSI bridge.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL153I: PARAMETERS CONFLICT, aaa = bbb ONLY ALLOWED IF ccc = ddd, aaa = eee FOR STATEMENT KEYWORD VALIDATION**

**Explanation:** Operand aaa = bbb can be coded only if operand ccc = ddd is specified.

**Utility Action:** Verify that operand bbb can be coded, this condition is required for validity of operand aaa. Value bbb is replaced by eee for statement keyword validation. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL154I: X25.MAXPIU VALUE MUST BE GREATER THAN OR EQUAL TO THE PRODUCT OF NCP BUFFER SIZE (BFRS KEYWORD ON BUILD) BY THE TRANSFR COUNT, MINUS 38.**

**Explanation:** This rule applies to the TRANSFR keyword coded on the X25.MCH or X25.LINE statement.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL155I: TRANSFR ONLY USEFULL WITH LLC TYPE 3, IGNORED.**

**Explanation:** The bbb keyword has been coded on virtual circuits of LLC type x; if the bbb keyword is active only with LLC type 3, then bbb is meaningless on this virtual circuit (the TRANSFR keyword is active only with LLC type 3).

**Utility Action:** Verify your request.

**Programmer Action:** This in an informational message.

---

**BAL156I: IN THE PREVIOUS MCH (ADDR=aaa), bbb IS NOT USEFUL SINCE LLCx IS NEITHER SUPPORTED IN THE LLCLIST FOR SVC(S) NOR SUPPORTED BY ANY PVC(S) (LLC KEYWORD).**

**Explanation:** The bbb keyword has been coded on virtual circuits of LLC type x; if no virtual circuit of this type is defined under the previous MCH aaa involved (neither PVC or SVC), then this keyword is meaningless (the TRANSFR keyword is active only with LLC type 3).

**Utility Action:** Verify your request.

**Programmer Action:** This is an informational message.

---

**BAL157I: IN THE PREVIOUS MCH (ADDR=aaa), bbb IS NOT USEFUL SINCE LLCx IS NOT SUPPORTED IN THE LLCLIST FOR SVC(S).**

**Explanation:** If only SVCs are coded under the MCH aaa, then the LLC type x needed by bbb must be included in the LLCLIST. The bbb keyword is meaningless (the TRANSFR keyword is active only with LLC type 3).

**Utility Action:** Verify your request.

**Programmer Action:** This is an informational message.

---

**BAL158I: IN THE PREVIOUS MCH (ADDR=aaa), bbb IS NOT USEFUL SINCE LLCx IS NOT SUPPORTED BY ANY PVC(S) (LLC KEYWORD).**

**Explanation:** If only PVCs are coded under the MCH aaa, then the LLC type needed by bbb must be supported by at least one PVC. The bbb keyword is meaningless (the TRANSFR keyword is active only with LLC type 3).

**Utility Action:** Verify your request.

**Programmer Action:** This is an informational message.

---

**BAL159I: INVALID DEFINITION, A SVC (TYPE=S) CANNOT BE CONNECTED TO A SWITCHED SUBAREA NODE (DSTNODE=INN) WHILE NO SVCSC HAS BEEN DEFINED ON THIS MCH BY SVCINN KEYWORD, DSTNODE=BNN ASSUMED FOR STATEMENT VALIDATION.**

**Explanation:** DSTNODE=INN is coded on the X25.LINE statement:

1. Either you have defined a normal switched virtual circuit TYPE=S, in that case DSTNODE=INN is not valid.
2. Or this virtual circuit is dedicated to a switched subarea node, in that case SVCINN=x is missing on the X25.MCH statement.

**Utility Action:** Verify your request. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL160I aaa NOT OPERATIVE WHEN bbb ON ccc, IGNORED.**

**Explanation:** The keyword aaa has no action when the bbb keyword is coded on the ccc statement. Example: DCI=YES is ignored when CAUSE=1984 is coded on X25.NET.

**Utility Action:** The keyword aaa is ignored. The generation process continues the validation of the generation definition.

**Programmer Action:** This is an informational message.

---

**BAL161I PARAMETERS CONFLICT, aaa = bbb IS NOT SUPPORTED ON ccc.**

**Explanation:** The keyword aaa=bbb is not allowed on a statement defining the ccc specific function. Example: INTWLINK keyword is not allowed on an X25.MCH dedicated to FTPI.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL162I PARAMETERS CONFLICT, aaa = bbb IS NOT ALLOWED ON ccc.**

**Explanation:** The keyword aaa=bbb is not compatible with the other ccc definition. Example: INTWLINK keyword is not allowed, regardless of whether LLC4 is defined on LLCLIST.

**Utility Action:** The keyword aaa is ignored. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL163I PARAMETERS CONFLICT, aaa = bbb IS REQUIRED ON ccc.**

**Explanation:** The keyword aaa=bbb is missing; it is required when the ccc definition is coded. Example: SPEED=6400 is required when the interworking function is coded on a MCH.

**Utility Action:** Add the missing keyword. The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL164I PARAMETERS CONFLICT, THIS MCH IS DEDICATED TO INTERWORKING CONNECTIONS FOR DIAL OUT BY THE PSDN, ONLY 1 SVC IS ALLOWED, REQUIRED.**

**Explanation:** More than one SVC is coded. However, only one SVC is allowed for the interworking connection for the dial-out-by-the-PSDN.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL165I PARAMETERS CONFLICT, MULTIPLE LUS ARE NOT ALLOWED ON PCNE CONNECTION, ONLY ONE X25.LU STATEMENT REQUIRED.**

**Explanation:** More than one LU is coded. However, only one LU is allowed for PCNE connection.

**Utility Action:** The generation process ends after validation of the generation definition.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL168I BALANPM ROUTINE NAME REMOVED - INVALID ON UACCTNG KEYWORD WHEN ALL THE X25.MCH CODED IN THE GENERATION HAVE ACCOUNT=NO.**

**Explanation:** BALANPM subvalue is removed from the UACCTNG keyword on the GENEND statement when the subvalue is not necessary.

**Utility Action:** The BALANPM keyword is ignored.

**Programmer Action:** This is an informational message.

---

**BAL184I IN THE PREVIOUS MCH (ADDR=aaa), GATE SUPPORT DEFINED BUT LLC4 IS NEITHER SUPPORTED IN LLCLIST FOR SVC(S) NOR SUPPORTED BY ANY PVC(S) (LLC KEYWORD).**

**Explanation:** The user has coded GATE=GENERAL, SUBADDR=NO on the X25.MCH statement, and all PVCs and SVCs coded are not eligible for LLC type 4.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL185I IN THE PREVIOUS MCH (ADDR=aaa), GATE SUPPORT DEFINED BUT LLC4 IS NOT SUPPORTED BY ANY PVC(S) (LLC KEYWORD).**

**Explanation:** The user has coded GATE=GENERAL, SUBADDR=NO on the X25.MCH statement, and all PVCs (no SVC defined on this MCH) coded are not eligible for LLC type 4.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL186I IN THE PREVIOUS MCH (ADDR=aaa), GATE SUPPORT DEFINED BUT LLC4 IS NOT SUPPORTED IN LLCLIST FOR SVC(S).**

**Explanation:** The user has coded GATE=GENERAL, SUBADDR=NO on the X25.MCH statement, and all SVCs (no PVC defined on this MCH) coded are not eligible for LLC type 4.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL187I SUBADDRESSING WILL USE 2 DIGITS ON THIS MCH.**

**Explanation:** The user coded 2-digit values on the SUBD keyword.

**Utility Action:** The last 2 digits of the called DTE address are used to select the CTCP.

**Programmer Action:** This is an informational message.

---

**BAL188I TYPE=S INVALID, WHEN THE FTPI PVC SUPPORT IS SELECTED ON X.25 MCH. THIS MCH MUST CONTAIN ONLY VIRTUAL CIRCUITS ON LEASED MODE AN LLC4 TYPE.**

**Explanation:** The user coded a switched virtual circuit for an MCH with FTPI=PVC. FTPI=PVC defines an FTPI MCH with PVCs only.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL194I CTCP x IS AN IMPLIED CTCP WHICH WAS NOT DEFINED IN THE CTCP STATEMENT OR THE AEFCTCP STATEMENT. AN ENTRY IN EITHER OF THOSE STATEMENTS MAY BE MISSING.**

**Explanation:** The user coded the CTCP and/or AEFCTCP keyword which resulted in CTCP x not getting defined.

**Utility Action:** The LU and SLUB will be added for the CTCP that was not coded. The generation process continues the validation of the generation definition.

**Programmer Action:** This is an informational message.

---

**BAL195I ADDRESS=xxx INVALID, X21 SWITCHED ACCESS TO ISDN THRU 3745-900 IS ONLY SUPPORTED ON LIC TYPE 12 ATTACHED TO A CLP.**

**Explanation:** The ADDRESS coded is not in the range of LIC 12 addresses.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL196I SPEED, NOT USER CODED, IS DEFAULTED TO 9600. VERIFY THAT THIS VALUE IS ACCURATE TO ENSURE CORRECT UTILIZATION OF THE ODLC LINE.**

**Explanation:** It is likely that the ODLC line supports a higher speed.

**Utility Action:** The speed for this line is defaulted to 9600.

**Programmer Action:** This is an informational message.

---

**BAL197I LIMIT EXCEEDED, aaa X.25 MODULO 128 LINES ARE DEFINED ON CLP POSITION bbb.**

**Explanation:** The X.25 modulo 128 line capacity of the CLP has been exceeded.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the generation and resubmit the job.

---

**BAL198I THE NUMBER OF X25.MCH STATEMENTS USING FAST CONNECT CANNOT EXCEED 36 AND ANY X25.MCH STATEMENT WHICH USES FAST CONNECT MUST BE CODED WITHIN THE FIRST 36 X25.MCH STATEMENTS.**

**Explanation:** The number of X25.MCH statements has exceeded 36.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the generation and resubmit the job.

---

**BAL199I aaa KEYWORD NOT SUPPORTED WHEN ODLC=YES; bbb.**

**Explanation:** The keyword is not supported on X.25 MCHs using X.25 ODLC.

**Utility Action:** The keyword is ignored.

**Programmer Action:** This is an informational message.

---

**BAL200I AN ERROR HAS OCCURED WHILE PROCESSING THE PRECEDING aaa CHANNEL(S) FOR X.25 ODLC SUPPORT; THE PROCESSING OF THE REMAINING LOGICAL CHANNEL(S) IS INTERRUPTED.**

**Explanation:** X.25 ODLC enforces the CCITT recommendation for the order of logical channels. There are four groups of logical channels that must be in the following order: PVCs, Call in SVCs, Two way SVCs, and Call SVCs. Within each group, logical channel numbers (LCNs) must be used consecutively. A logical channel was found that did not conform to the CCITT recommendation.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL201I LCGN=aaa INVALID; NO DISCONTINUITY IS ALLOWED IN THE RANGE OF THE LCGN NUMBERS FOR X.25 ODLC SUPPORT. LCGN=bbb IS EXPECTED; REQUIRED.**

**Explanation:** X.25 ODLC enforces the CCITT recommendation for the order of logical channels. There are four groups of logical channels that must be in the following order: PVCs, Call in SVCs, Two way SVCs and Call out SVCs. Within each group, logical channel numbers (LCNs) must be used consecutively. A logical channel group was found that did not conform to the CCITT recommendation.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL202I ERROR, THIS aaa DEFINITION DOES NOT FOLLOW THE CCITT RULE OF LOGICAL CHANNEL RANGES. THE LOGICAL CHANNELS MUST BE ASSIGNED IN THE FOLLOWING ORDER: PVC GROUP - ONE WAY INCOMING SVCS - TWO-WAYS SVCS - ONE WAY OUTGOING SVCS; REQUIRED.**

**Explanation:** X.25 ODLC enforces the CCITT recommendation for the order of logical channels. There are four groups of logical channels that must be in the following order: PVCs, Call in SVCs, Two way SVCs, and Call out SVCs. Within each group, logical channel numbers (LCNs) must be used consecutively. A logical channel was found that did not conform to the CCITT recommendation.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL203I LCN=aaa INVALID; NO DISCONTINUITY IS ALLOWED IN THE RANGE OF THE LCN NUMBERS FOR X.25 ODLC SUPPORT. LCN=bbb IS EXPECTED; REQUIRED.**

**Explanation:** X.25 ODLC enforces the CCITT recommendation for the order of logical channels. There are four groups of logical channels that must be in the following order: PVCs, Call in SVCs, Two way SVCs, and Call out SVCs. Within each group, logical channel numbers (LCNs) must be used consecutively. A logical channel was found that did not conform to the CCITT recommendation.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL204I ERROR. LOGICAL CHANNELS NUMBERS CONTINUITY IS DISRUPTED WITH LCGN=/S; NO DISCONTINUITY IS AUTHORIZED IN THE RANGE OF THE SWITCHED LOGICAL CHANNELS DEFINITION FOR X.25 ODLC SUPPORT; REQUIRED.**

**Explanation:** X.25 ODLC enforces the CCITT recommendation for the order of logical channels. There are four groups of logical channels that must be in the following order: PVCs, Call in SVCs, Two way SVCs, and Call out SVCs. Within each group, logical channel numbers (LCNs) must be used consecutively. A logical channel group was found that did not conform to the CCITT recommendation.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.



---

**BAL206I aaa=bbb INVALID. ONLY ccc AUTHORIZED FOR ddd. REQUIRED.**

**Explanation:** For the ddd function, ccc is the only valid value for keyword aaa.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL207I aaa=bbb INVALID. ONLY ccc IS SUPPORTED BY ddd, THEN ONLY eee ALLOWED.**

**Explanation:** aaa=eee must be coded because ccc is required by the ddd function.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL208I aaa=bbb REPLACED BY ccc. ddd CAN ONLY TAKE VALUE eee FOR NETWORK PERFORMANCE MONITORING FOR X.25 ODLC SUPPORT.**

**Explanation:** X.25 ODLC does not allow selection of resources on which to collect NPM accounting data by LLC.

**Utility Action:** Accounting data is collected on all resources on this MCH.

**Programmer Action:** This is an informational message.

---

**BAL209I ALL THE X25.MCH ARE DEDICATED TO X.25 ODLC SUPPORT; THEREFORE aaa IS NOT ALLOWED IN THIS GENERATION DECK.**

**Explanation:** X.25 ODLC does not support DATE or PAD.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL210I PARAMETERS CONFLICT, aaa IS NOT A RELATIVE LINE NUMBER FOR bbb.**

**Explanation:** A valid CLP line number must be specified for an X.25 ODLC MCH.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the error and resubmit the job.

---

**BAL211I aaa=bbb INVALID. ALL THE SWITCHED VIRTUAL CIRCUITS DEFINED UNDER THE SAME X25.MCH STATEMENT MUST POINT TO THE SAME CORRESPONDING VCCPT**

**Explanation:** The VCCINDX of all of the X25.VC or X25.LINE statements for switched lines must point to the same X25.VCCPT statement.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the generation and resubmit the job.

---

---

**BAL222I aaa IS A bbb KEYWORD WHICH IS NOT ALLOWED ON THE ccc STATEMENT. IGNORED FOR KEYWORD VALIDATION.**

**Explanation:** The aaa keyword is a type (aaa) of keyword that is not allowed on the ccc statement.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the generation and resubmit the job.

---

**BAL223I THE CORRECT FORMAT FOR THE aaa KEYWORD IS “xxx,ccc,vvv”, WHERE xxx IS GROUP, LINE, PU, LU, OR ALL, ccc IS THE bbb KEYWORD NAME, AND vvv IS THE KEYWORD VALUE.**

**Explanation:** The aaa keyword is not specified in the correct format.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the generation and resubmit the job.

---

**BAL224I THE CORRECT FORMAT FOR THE aaa KEYWORD IS “xxx,yyy”, WHERE xxx IS GROUP, LINE, PU, LU, OR ALL, AND yyy IS THE VALUE. xxx IS INVALID.**

**Explanation:** The aaa keyword is not specified in the correct format. The xxx value was not GROUP, LINE, PU, LU, or ALL.

**Utility Action:** The generation process ends after validation of the generation definitions.

**Programmer Action:** Correct the generation and resubmit the job.

---

## List of Abbreviations

|                |                                                                                |              |                                                         |
|----------------|--------------------------------------------------------------------------------|--------------|---------------------------------------------------------|
| <b>AEF</b>     | address extension facility                                                     | <b>LLC</b>   | logical link control                                    |
| <b>ASCII</b>   | American National Standard Code for Information Interchange                    | <b>LU</b>    | logical unit                                            |
| <b>AU</b>      | access unit                                                                    | <b>LUSIM</b> | LU simulator                                            |
| <b>BNN</b>     | boundary network node                                                          | <b>MCH</b>   | multichannel link                                       |
| <b>BSC</b>     | binary synchronous communication                                               | <b>MIC</b>   | middle-in-chain                                         |
| <b>CCITT</b>   | Consultative Committee on International Telegraph and Telephone                | <b>MVS</b>   | Multiple Virtual Storage                                |
| <b>CCU</b>     | central control unit                                                           | <b>NCP</b>   | Network Control Program                                 |
| <b>CD-ROM</b>  | compact disk read-only memory                                                  | <b>NDF</b>   | NCP/EP Definition Facility                              |
| <b>CEN</b>     | composite end node                                                             | <b>NPI</b>   | numbering plan identifier                               |
| <b>CICS</b>    | Customer Information Control System                                            | <b>NPM</b>   | NetView Performance Monitor                             |
| <b>CLP</b>     | communication line processor                                                   | <b>NPSI</b>  | X.25 Network Control Program Packet Switching Interface |
| <b>CP</b>      | control point                                                                  | <b>NRF</b>   | Network Routing Facility                                |
| <b>CPS</b>     | complete packet sequence                                                       | <b>ODLC</b>  | outbound data link control                              |
| <b>CPS/MBS</b> | complete packet sequence M-bit sequence                                        | <b>OIC</b>   | only-in-chain                                           |
| <b>CSS</b>     | connectivity subsystem                                                         | <b>OSI</b>   | Open Systems Interconnection                            |
| <b>CTCP</b>    | communication and transmission control program                                 | <b>OUFT</b>  | optional user facility table                            |
| <b>CUD</b>     | call user data                                                                 | <b>PAD</b>   | packet assembler/disassembler                           |
| <b>CUG</b>     | closed user group                                                              | <b>PCNE</b>  | protocol converter for non-SNA equipment                |
| <b>DATE</b>    | dedicated access to X.25 transport extension                                   | <b>PIU</b>   | path information unit                                   |
| <b>DCE</b>     | data circuit-terminating equipment                                             | <b>PRPQ</b>  | programming request for price quotation                 |
| <b>DM</b>      | disconnect mode                                                                | <b>PSDN</b>  | packet switching data network                           |
| <b>DSC</b>     | data stream compatibility                                                      | <b>PSH</b>   | physical services header                                |
| <b>DSR</b>     | data set ready                                                                 | <b>PTF</b>   | program temporary fix                                   |
| <b>DTE</b>     | data terminal equipment                                                        | <b>PU</b>    | physical unit                                           |
| <b>DTR</b>     | data terminal ready                                                            | <b>PVC</b>   | permanent virtual circuit                               |
| <b>ENP</b>     | enable presentation (characters)                                               | <b>QLLC</b>  | qualified logical link control                          |
| <b>FC</b>      | fast connect                                                                   | <b>QOS</b>   | quality of service                                      |
| <b>FIC</b>     | first-in-chain                                                                 | <b>RR</b>    | receive ready (packet)                                  |
| <b>FTPI</b>    | Fast Transaction Processing Interface                                          | <b>RU</b>    | request/response unit                                   |
| <b>GATE</b>    | general access to X.25 transport extension                                     | <b>SCS</b>   | SNA character string                                    |
| <b>GTMOSI</b>  | General Teleprocessing Monitor for Open Systems Interconnection                | <b>SDLC</b>  | Synchronous Data Link Control                           |
| <b>HTPSS</b>   | high performance transmission subsystem                                        | <b>SHM</b>   | short hold mode                                         |
| <b>IAU</b>     | interworking access units                                                      | <b>SLR</b>   | service level reporter                                  |
| <b>IDBLK</b>   | identification block                                                           | <b>SLU</b>   | secondary logical unit                                  |
| <b>IMS</b>     | Information Management System                                                  | <b>SMN</b>   | switched major node                                     |
| <b>INOP</b>    | inoperative                                                                    | <b>SMP</b>   | System Modification Program                             |
| <b>INP</b>     | inhibit presentation (characters)                                              | <b>SNA</b>   | Systems Network Architecture                            |
| <b>IPL</b>     | initial program load                                                           | <b>SNI</b>   | SNA network interconnection                             |
| <b>ISDN</b>    | integrated services digital network                                            | <b>SS</b>    | start-stop                                              |
| <b>ISO</b>     | International Organization for Standardization                                 | <b>SSCP</b>  | system services control point                           |
| <b>ITU-TS</b>  | International Telecommunication Union-Telecommunication Standardization Sector | <b>SVC</b>   | switched virtual circuit                                |
| <b>LAPB</b>    | link access protocol balanced                                                  | <b>SVCSC</b> | switched virtual circuit subarea communication          |
| <b>LCG</b>     | logical channel group                                                          | <b>TA</b>    | terminal adapter                                        |
| <b>LCGN</b>    | logical channel group number                                                   | <b>TOA</b>   | type of address                                         |
| <b>LCN</b>     | logical channel number                                                         | <b>TSO</b>   | Time Sharing Option                                     |
| <b>LIC</b>     | line interface coupler                                                         | <b>TSS</b>   | transmission subsystem                                  |
| <b>LIC</b>     | last-in-chain                                                                  | <b>VM</b>    | Virtual Machine                                         |
| <b>LKEDT</b>   | link edit                                                                      | <b>VRPWS</b> | virtual route pacing window size                        |
|                |                                                                                | <b>VSE</b>   | Virtual Storage Extended                                |
|                |                                                                                | <b>VTAM</b>  | Virtual Telecommunications Access Method                |



---

# Glossary

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology*. Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The Network Working Group Request for Comments: 1208.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

The following cross-references are used in this glossary:

**Contrast with:** This refers to a term that has an opposed or substantively different meaning.

**Synonym for:** This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

**Synonymous with:** This is a backward reference from a defined term to all other terms that have the same meaning.

**See:** This refers the reader to multiple-word terms that have the same last word.

**See also:** This refers the reader to terms that have a related, but not synonymous, meaning.

**Deprecated term for:** This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

## A

**abend.** (1) Abnormal end of task; the termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing. (2) Synonym for *abnormal termination*.

**abnormal termination.** (1) The cessation of processing prior to planned termination. (T) (2) A system failure or operator action that causes a job to end unsuccessfully. (3) Synonymous with *abend*.

**ACB.** (1) In VTAM, access method control block. (2) In NCP, adapter control block. (3) Application control block.

**accept.** (1) In a VTAM application program, to establish a session with a logical unit (LU) in response to a CINIT request from a system services control point (SSCP). The session-initiation request may begin when a terminal user logs on, a VTAM application program issues a macroinstruction, or a VTAM operator issues a command. See also *acquire*. (2) An SMP process that moves distributed code and MVS-type programs to the distribution libraries.

**ACCESS.** In the Simple Network Management Protocol (SNMP), the clause in a Management Information Base (MIB) module that defines the minimum level of support that a managed node provides for an object.

**access method.** (1) A technique, implemented in software, that controls the flow of information through a network. (2) A technique for moving data between main storage and input/output devices.

**access method control block (ACB).** A control block that links an application program to VSAM or VTAM.

**access procedure.** The procedure or protocol used to gain access to a shared resource; for example, in a local area network, the shared resource is the transmission medium. The medium access procedures specified by the IEEE 802 standard are CSMA/CD token, bus, and ring.

**access unit.** A unit that allows attaching devices to access a local area network (LAN) at a central point, such as a wiring closet or an open work area.

**ACF/NCP.** Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

## Glossary

**ACF/TCAM.** Advanced Communications Function for the Telecommunications Access Method. Synonym for *TCAM*.

**ACF/VTAM.** Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

**acknowledgment.** (1) The transmission, by a receiver, of acknowledge characters as an affirmative response to a sender. (T) (2) An indication that an item sent was received.

**acquire.** (1) In VTAM, to take over resources that were formerly controlled by an access method in another domain or to resume control of resources that were controlled by that domain but released. Contrast with *release*. See also *resource takeover*. (2) In a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macroinstruction. See also *accept*.

**action.** (1) In the AIX operating system, a defined task that an application performs. An action modifies the properties of an object or manipulates the object in some way. (2) An operation on a managed object, the semantics of which are defined as part of the managed object class definition.

**activate.** To make a resource ready to perform its function. Contrast with *deactivate*.

**active.** (1) Operational. (2) Pertaining to a node or device that is connected or is available for connection to another node or device. (3) The state of a resource when it has been activated and is operational. (4) In the AIX operating system, pertaining to the window pane in which the text cursor is currently positioned. (5) In VTAM, pertaining to a major or minor node that has been activated by VTAM. Most resources are activated as part of VTAM start processing or as the result of a VARY ACT command. (6) Contrast with *inactive* and *inoperative*. See also *pending active session*.

**ACTLINK.** Activate link.

**ACTLU.** Activate logical unit. In SNA, a command used to start a session on a logical unit.

**ACTPU.** Activate physical unit. In SNA, a command used to start a session on a physical unit.

**adapter.** A part that electrically or physically connects a device to a computer or to another device.

**adapter control block (ACB).** In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

**adaptive pacing.** Synonym for *adaptive session-level pacing*.

**adaptive session-level pacing.** A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission within a network to adapt dynamically to variations in availability and demand of buffers on a session-by-session basis. Session-level pacing occurs within independent stages along the session path according to local congestion at the intermediate and endpoint nodes. Synonymous with *adaptive pacing* and *adaptive session pacing*. Contrast with *fixed session-level pacing*.

**adaptive session pacing.** Synonym for *adaptive session-level pacing*.

**address.** In data communication, the unique code assigned to each device or workstation connected to a network.

**addressing.** (1) The assignment of addresses to the instructions of a program. (2) A means of identifying storage locations. (3) In data communication, the way in which a station selects the station to which it is to send data. (4) Specifying an address or location within a file.

**adjacent link station (ALS).** (1) In SNA, a link station directly connected to a given node by a link connection over which network traffic can be carried.

**Note:** Several secondary link stations that share a link connection do not exchange data with each other and therefore are not adjacent to each other.

(2) With respect to a specific node, a link station partner in an adjacent node.

**Advanced Peer-to-Peer Networking (APPN).** An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; (b) dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; (c) dynamic definition of network resources; and (d) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control and supports multiple LU types, including LU 2, LU 3, and LU 6.2.

**Advanced Peer-to-Peer Networking (APPN) end node.** A node that provides a broad range of end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node. It uses these sessions to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests,

and to obtain management services. An APPN end node can also attach to a subarea network as a peripheral node or to other end nodes.

**Advanced Peer-to-Peer Networking (APPN) network.** A collection of interconnected network nodes and their client end nodes.

**Advanced Peer-to-Peer Networking (APPN) network node.** A node that offers a broad range of end-user services and that can provide the following:

- Distributed directory services, including registration of its domain resources to a central directory server
- Topology database exchanges with other APPN network nodes, enabling network nodes throughout the network to select optimal routes for LU-LU sessions based on requested classes of service
- Session services for its local LUs and client end nodes
- Intermediate routing services within an APPN network

**Advanced Peer-to-Peer Networking (APPN) node.** An APPN network node or an APPN end node.

**AID.** Attention identifier.

**alert.** (1) A message sent to a management services focal point in a network to identify a problem or an impending problem. (2) In the NetView and NETCENTER programs, a high priority event that warrants immediate attention.

**ALS.** Adjacent link station.

**analog.** (1) Pertaining to data consisting of continuously variable physical quantities. (A) (2) Contrast with *digital*.

**AND operation.** Synonym for *conjunction*.

**APPL.** Application program.

**application.** A collection of software components used to perform specific types of user-oriented work on a computer.

**application control block (ACB).** The control blocks created from the output of DBDGEN and PSBGEN and placed in the ACB library for use during online and DBB region type execution of IMS/VS.

**application program.** (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**apply.** An SMP process that moves distributed code and MVS-type programs to the system libraries.

**Apply.** A push button that carries out the selected choices in a window without closing the window.

**APPN.** Advanced Peer-to-Peer Networking.

**area.** In Internet and DECnet routing protocols, a subset of a network or gateway grouped together by definition of the network administrator. Each area is self-contained; knowledge of an area's topology remains hidden from other areas.

**ASCII (American National Standard Code for Information Interchange).** The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

**asynchronous (ASYN).** (1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions.

**asynchronous balanced mode (ABM).** An operational mode of a balanced data link in which either combined station can send commands at any time and can initiate transmission of response frames without explicit permission from the other combined station.

**attach.** (1) In programming, to create a task that can be executed asynchronously with the execution of the mainline code. (2) To connect a device logically to a ring network.

**attention identifier (AID).** A character in a data stream indicating that the user has pressed a key, such as the Enter key, that requests an action by the system.

**autologon.** Synonym for *automatic logon*.

**automatic activation.** In VTAM, the activation of links and link stations in adjacent subarea nodes as a result of channel device name or RNAME specifications related to an activation command that names a subarea node. See also *direct activation*.

**automatic logon.** (1) A process by which VTAM automatically creates a session-initiation request to establish a session between two logical units (LUs). The session is between a designated primary logical unit (PLU) and a secondary logical unit (SLU) that is neither queued for nor in session with another PLU. Synonymous with *autologon*. See also *controlling application program* and *controlling logical unit*. (2) In VM, a process by

## Glossary

which a virtual machine is initiated by other than the user of that virtual machine; for example, the primary VM operator's virtual machine is activated automatically during VM initialization.

**autotask.** An unattended NetView operator station task that does not require a terminal or a logged-on user. Autotasks can run independently of VTAM and are typically used for automated console operations. Contrast with *logged-on operator*.

**available.** In VTAM, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

## B

**B-channel.** In the integrated services digital network (ISDN), a 64 kilobits-per-second channel for the transport of speech or data between the ISDN service provider and user.

**back-level.** Pertaining to an earlier release of an IBM product, which may not support a particular, current function.

**baseband.** (1) A frequency band occupied by a signal, or by a number of multiplexed signals. (T) (2) A frequency band that uses the complete bandwidth of a transmission.

**batch.** (1) An accumulation of data to be processed. (2) A group of records or data processing jobs brought together for processing or transmission. (3) Pertaining to activity involving little or no user action. Contrast with *interactive*.

**begin bracket.** In SNA, the value (binary 1) of the begin-bracket indicator in the request header (RH) of the first request in the first chain of a bracket; the value denotes the start of a bracket. Contrast with *end bracket*. See also *bracket*.

**behavior.** The way in which managed objects, name bindings, attributes, notifications, and operations interact with the actual resources that they model and with each other.

**bilateral closed user group (BCUG).** (1) In data communication, two users who have bilaterally agreed to communicate with each other, but not with other users. See *closed user group (CUG)*.

**Note:** Each user can belong to more than one bilateral closed user group and to more than one closed user group by means of outgoing access.

(2) In X.25 communications, an optional facility that allows calls to be made only between two designated DTEs.

**binary synchronous communication (BSC).** (1) A form of telecommunication line control that uses a standard set of transmission control characters and control character sequences, for binary synchronous transmission of binary-coded data between stations. (2) Contrast with *Synchronous Data Link Control (SDLC)*.

**BIND.** In SNA, a request to activate a session between two logical units (LUs). See also *session activation request*. Contrast with *UNBIND*.

**BIU segment.** In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either (a) a request/response header (RH) followed by all or a part of a request/response unit (RU) or (b) a part of an RU. Synonymous with *segment*.

**block.** A string of data elements recorded or transmitted as a unit. The elements may be characters, words, or physical records. (T)

**BN.** Boundary node.

**BNN.** Boundary network node.

**boundary function (BF).** (1) In SNA, a capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) In SNA, the component that provides these capabilities.

**boundary network node (BNN).** (1) In SNA, deprecated term for *boundary node (BN)*. (2) In NCP, deprecated term for *peripheral node*.

**boundary node (BN).** In SNA, a subarea node with boundary function.

**Note:** A subarea node may be a boundary node, an intermediate routing node, both, or neither, depending on how it is used in the network.

**bps.** Bits per second.

**bracket.** In SNA, one or more chains of request units and their responses that are exchanged between two session partners and that represent a transaction between them. A bracket must be completed before another bracket can be started. Examples of brackets are database inquiries/replies, update transactions, and remote job entry output sequences to workstations.

**bracket protocol.** In SNA, a data flow control protocol in which exchanges between two session partners are achieved through the use of brackets, with one partner designated at session activation as the first speaker and



the other as the bidder. The bracket protocol involves bracket initiation and termination rules.

**bridge.** (1) A functional unit that interconnects two local area networks that use the same logical link control protocol but may use different medium access control protocols. (T) (2) A functional unit that interconnects multiple LANs (locally or remotely) that use the same logical link control protocol but that can use different medium access control protocols. A bridge forwards a frame to another bridge based on the medium access control (MAC) address. (3) In the connection of local loops, channels, or rings, the equipment and techniques used to match circuits and to facilitate accurate data transmission. (4) Contrast with *gateway* and *router*.

**broadband.** (1) A frequency band broad enough to be divided into several narrower bands, each of which can be used for different purposes or be made available to different users. Synonymous with *wideband*. (T) (2) A frequency band divisible into several narrower bands so that different kinds of transmission (such as voice, video, and data) can occur at the same time. Synonymous with *wideband*. See also *baseband*. (3) Transmission media and techniques that use a broad frequency range, divided into sub-bands of narrower frequency, so that different kinds of transmission can occur at the same time.

**browse.** (1) To look at records in a file. (2) In the NetView Graphic Monitor Facility, to open a view that cannot receive status changes from the NetView program. Contrast with *monitor*.

**BSC.** Binary synchronous communication.

**buffer.** (1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (A) (2) To allocate and schedule the use of buffers. (A) (3) A portion of storage used to hold input or output data temporarily.

**bypass.** To eliminate a station or an access unit from a ring network by allowing the data to flow in a path around it.

**byte.** (1) A string that consists of a number of bits, treated as a unit, and representing a character. (T) (2) A binary character operated upon as a unit and usually shorter than a computer word. (A) (3) A group of 8 adjacent binary digits that represent one EBCDIC character.

## C

**CA.** (1) Channel adapter. (2) Channel attachment.

**call.** (1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (I) (A) (2) In data communication, the actions necessary to make a connection between two stations on a switched line. (3) In communications, a conversation between two users. (4) To transfer control to a procedure, program, routine, or subroutine. (5) To attempt to contact a user, regardless of whether the attempt is successful.

**call-accepted packet.** A call supervision packet that a called data terminal equipment (DTE) transmits to indicate to the data circuit-terminating equipment (DCE) that it accepts the incoming call.

**call connected packet.** A call supervision packet that a data circuit-terminating equipment (DCE) transmits to indicate to a calling data terminal equipment (DTE) that the connection for the call has been completely established.

**call establishment.** The complete sequence of events necessary to establish a data connection.

**call progress signal.** A call control signal transmitted from the data circuit-terminating equipment (DCE) to the calling data terminal equipment (DTE) to indicate the progress of the establishment of a call, the reason why the connection could not be established, or any other network condition. (T)

**call request packet.** (1) A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network. (2) In X.25 communications, a call supervision packet transmitted by a DTE to ask for a call establishment through the network.

**call user data (CUD).** In X.25 communications, data optionally included in the call-request packet by the user application.

**called party.** On a switched line, the location to which a connection is established.

**CALLIN.** The logical channel type on which the data terminal equipment (DTE) can receive a call, but cannot send one.

**calling.** (1) The process of transmitting selection signals in order to establish a connection between data stations. (I) (A) (2) In X.25 communications, pertaining to the location or user that makes a call.

**calling address.** See *network user address*.

## Glossary

**CALLOUT.** The logical channel type on which the data terminal equipment (DTE) can send a call, but cannot receive one.

**casual connection.** (1) In a subarea network, a connection in which type 5 nodes are attached through the boundary function using low-entry networking (LEN). Therefore, the nodes appear as LEN nodes rather than subarea nodes. (2) In an APPN network, a connection between an end node and a network node with different network identifiers.

**CCITT.** International Telegraph and Telephone Consultative Committee. This was an organization of the International Telecommunication Union (ITU). On 1 March 1993 the ITU was reorganized, and responsibilities for standardization were placed in a subordinate organization named the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-TS). "CCITT" continues to be used for recommendations that were approved before the reorganization.

**CCU.** Central control unit.

**CD.** Compact disc.

**CD-ROM.** High-capacity read-only memory in the form of an optically read compact disc.

**chain.** (1) A group of logically linked user data records processed by LU 6.2. (2) A group of request units delimited by begin-chain and end-chain. Responses are always single-unit chains. See *RU chain*.

**channel.** (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) In data communication, a means of one-way transmission. (3) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

**channel adapter.** A communication controller hardware unit that is used to attach the communication controller to a host channel.

**channel-attached.** (1) Pertaining to the attachment of devices directly by input/output channels to a host processor. (2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

**channel link.** A System/370 I/O channel to control unit interface that has an SNA network address. A channel link can be either a subarea link or a peripheral link and is defined in an NCP generation definition using the GROUP, LINE, and PU definition statements.

**character-coded.** Synonym for *unformatted*.

**CICS.** Customer Information Control System.

**circuit.** (1) One or more conductors through which an electric current can flow. See *physical circuit* and *virtual circuit*. (2) A logic device.

**circuit switching.** (1) A process that, on demand, connects two or more data terminal equipment (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (I) (A) (2) Synonymous with *line switching*. (3) See also *message switching* and *packet switching*.

**class.** In object-oriented design or programming, a group of objects that share a common definition and that therefore share common properties, operations, and behavior. Members of the group are called instances of the class.

**class of service (COS).** A set of characteristics (such as route security, transmission priority, and bandwidth) used to construct a route between session partners. The class of service is derived from a mode name specified by the initiator of a session.

**clear confirmation packet.** See *DCE clear confirmation packet*.

**clear indication packet.** A call supervision packet that a data circuit-terminating equipment (DCE) transmits to inform a data terminal equipment (DTE) that a call has been cleared.

**clear request packet.** A call supervision packet transmitted by a data terminal equipment (DTE) to ask that a call be cleared.

**clear to send (CTS).** In data communication, a signal raised by data circuit-terminating equipment (DCE) when it is ready to accept data, usually in response to request to send (RTS) being raised.

**clocking.** (1) In binary synchronous communication, the use of clock pulses to control synchronization of data and control characters. (2) A method of controlling the number of data bits sent on a telecommunication line in a given time.

**closed user group (CUG).** In data communication, a group of users who can communicate with other users in the group, but not with users outside the group.

**Note:** A data terminal equipment (DTE) may belong to more than one closed user group.

**closed user group with outgoing access.** (1) A closed user group that has a user-assigned facility which enables that user to communicate with other users of a data network transmission service where appropriate, with users having a data terminal equipment connected to any other switched network to which

interworking facilities are available, or both. (T) (A) (2) In data communication, a closed user group in which one or more users can communicate with users outside the closed group, under certain conditions.

**CLSDST.** Close destination.

**cluster.** (1) A station that consists of a control unit (a cluster controller) and the terminals attached to it. (2) A group of APPN nodes that have the same network ID and the same topology database. A cluster is a subset of a NETID subnetwork.

**cluster controller.** A device that can control the input/output operations of more than one device connected to it. A cluster controller may be controlled by a program stored and executed in the unit; for example, the IBM 3601 Finance Communication Controller. Or, it may be entirely controlled by hardware; for example, the IBM 3272 Control Unit. See also *cluster*.

**cold start.** (1) The start of a database management system without preprocessing before-images or after-images. (T) (2) A system start, using an initial program load procedure.

**collision.** An unwanted condition that results from concurrent transmissions on a channel. (T)

**command.** (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in an FID4 transmission header. (3) See also *VTAM operator command*.

**common operations services (COS).** The portion of SNA management services that pertains to the major vectors for limited remote operations control.

**common transport semantics (CTS).** The layer of the Networking Blueprint above the transport layer that makes the services of transport providers available to the transport user. See also *transport-layer protocol boundary (TLPB)*.

**Common User Access (CUA) architecture.** Guidelines for the dialog between a human and a workstation or terminal.

**communication controller.** A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit. It manages the details of line control and the routing of data through a network.

**compact disc (CD).** (1) A disc, usually 4.75 inches in diameter, from which data is read optically by means of a laser. (2) A disc with information stored in the form of pits along a spiral track. The information is decoded by a compact-disc player and interpreted as digital audio data, which most computers can process.

**Communication Link Process (CLP).** (1) A 3746-900 adapter for a TP link.

**component.** (1) Hardware or software that is part of a functional unit. (2) A part of a structured type or value, such as an array element or a record field.

**configuration.** (1) The manner in which the hardware and software of an information processing system are organized and interconnected. (T) (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

**configuration services.** One of the types of network services in a control point (SSCP, NNCP, or ENCP). Configuration services activates, deactivates, and records the status of physical units, links, and link stations.

**conjunction.** The Boolean operation whose result has the Boolean value 1 if and only if each operand has the Boolean value 1. (I) (A) Synonymous with *AND operation*.

**connected.** In VTAM, the state of a physical unit (PU) or a logical unit (LU) that has an active physical path to the host processor containing the system services control point (SSCP) that controls the respective PU or LU.

**connection.** (1) In data communication, an association established between functional units for conveying information. (I) (A) (2) In Open Systems Interconnection architecture, an association established by a given layer between two or more entities of the next higher layer for the purpose of data transfer. (T) (3) In VTAM, synonym for *physical connection*. (4) In SNA, the network path that links together two logical units (LUs) in different nodes to enable them to establish communications. (5) In X.25 communication, a virtual circuit between two data terminal equipments (DTEs). A switched virtual circuit (SVC) connection lasts for the duration of a call; a permanent virtual circuit (PVC) is a permanent connection between the DTEs. (6) In TCP/IP, the path between two protocol applications that provides reliable data stream delivery service. In Internet, a connection extends from a TCP application on one system to a TCP application on another system.

## Glossary

**connectivity.** (1) The capability of a system or device to be attached to other systems or devices without modification. (T) (2) The capability to attach a variety of functional units without modifying them.

**connectivity subsystem (CSS).** An expansion frame, such as the 3746 Model 900, that extends connectivity and enhances the performance of the IBM 3745 Communication Controller.

**constraints.** In NETDA/2, the set of essential requirements specified with the node, connection, or application definitions. A change in a constraint value changes the input to the network design. Contrast with *parameters*.

**contention.** In a session, a situation in which both NAUs attempt to initiate the same action at the same time, such as when both attempt to send data in a half-duplex protocol (half-duplex contention), or both attempt to start a bracket (bracket contention). At session initiation, one NAO is defined to be the contention winner; its action will take precedence when contention occurs. The contention loser must get explicit or implicit permission from the contention winner to begin its action.

**control block.** (1) A storage area used by a computer program to hold control information. (I) (2) In the IBM Token-Ring Network, a specifically formatted block of information provided from the application program to the Adapter Support Interface to request an operation.

**control character.** A character whose occurrence in a particular context specifies a control function. (T)

**control point (CP).** (1) A component of an APPN or LEN node that manages the resources of that node. In an APPN node, the CP is capable of engaging in CP-CP sessions with other APPN nodes. In an APPN network node, the CP also provides services to adjacent end nodes in the APPN network. (2) A component of a node that manages resources of that node and optionally provides services to other nodes in the network. Examples are a system services control point (SSCP) in a type 5 subarea node, a network node control point (NNCP) in an APPN network node, and an end node control point (ENCP) in an APPN or LEN end node. An SSCP and an NNCP can provide services to other nodes.

**control program.** (1) A computer program designed to schedule and to supervise the execution of programs of a computer system. (I) (A) (2) The part of the AIX Base Operating System that determines the order in which basic functions should be performed. (3) See *VM/370 control program (CP)*.

**control vector.** One of a general class of RU substructures that has variable length, is carried within

some enclosing structure, and has a one-byte key used as an identifier.

**controller.** A device that coordinates and controls the operation of one or more input/output devices, such as workstations, and synchronizes the operation of such devices with the operation of the system as a whole.

**controlling application program.** In VTAM, an application program with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. See also *automatic logon* and *controlling logical unit*.

**controlling logical unit.** In VTAM, a logical unit with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also *automatic logon* and *controlling application program*.

**conversation.** A logical connection between two transaction programs using an LU 6.2 session. Conversations are delimited by brackets to gain exclusive use of a session.

**Copy.** A choice that places a copy of a selected object onto the clipboard.

**COS.** (1) Class of service. (2) Common operations services.

**CP.** (1) Control point. (2) VM control program.

**CP name.** A network-qualified name of a control point (CP), consisting of a network ID qualifier identifying the network (or name space) to which the CP's node belongs, and a unique name within the scope of that network ID identifying the CP. Each APPN or LEN node has one CP name, assigned to it at system-definition time.

**CSS.** Connectivity subsystem.

**CTS.** (1) Common transport semantics. (2) Clear to send.

**CUA.** (1) Common User Access. (2) In VTAM, channel unit address.

**CUD.** Call user data.

**CUG.** Closed user group.

**Customer Information Control System (CICS).** An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

**CWALL.** An NCP threshold of buffer availability, below which the NCP will accept only high-priority path information units (PIUs).

**cylinder.** (1) In an assembly of magnetic disks, the set of all tracks that can be accessed by all the magnetic heads of a comb in a given position. (2) The tracks of a disk storage device that can be accessed without repositioning the access mechanism.

## D

**D-bit.** Delivery-confirmation bit. In X.25 communications, the bit in a data packet or call-request packet that is set to 1 if end-to-end acknowledgment (delivery confirmation) is required from the recipient.

**DACTLU.** Deactivate logical unit.

**data.** (1) A re-interpretable representation of information in a formalized manner suitable for communication, interpretation, or processing. Operations can be performed upon data by humans or by automatic means. (T) (2) Any representations such as characters or analog quantities to which meaning is or might be assigned. (A) (3) A representation of facts or instructions in a form suitable for communication, interpretation, or processing by human or automatic means. Data include constants, variables, arrays, and character strings.

**Note:** Programmers make a distinction between instructions and the data they operate on; however, in the usual sense of the word, data includes programs and program instructions.

**data circuit.** (1) A pair of associated transmit and receive channels that provide a means of two-way data communication. (l) (2) In SNA, synonym for *link connection*. (3) See also *physical circuit* and *virtual circuit*.

### Notes:

1. Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
2. Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

**data circuit-terminating equipment (DCE).** In a data station, the equipment that provides the signal conversion and coding between the data terminal equipment (DTE) and the line. (l)

### Notes:

1. The DCE may be separate equipment or an integral part of the DTE or of the intermediate equipment.
2. A DCE may perform other functions that are usually performed at the network end of the line.

**data communication.** (1) Transfer of data among functional units by means of data transmission according to a protocol. (T) (2) The transmission, reception, and validation of data. (A)

**data definition name (ddname).** The name of a data definition (DD) statement that corresponds to a data control block that contains the same name.

**data link.** In SNA, synonym for *link*.

**data link control (DLC).** A set of rules used by nodes on a data link (such as an SDLC link or a token ring) to accomplish an orderly exchange of information.

**data link control (DLC) layer.** In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

**Note:** The DLC layer is usually independent of the physical transport mechanism and ensures the integrity of data that reaches the higher layers.

**data link level.** (1) In the hierarchical structure of a data station, the conceptual level of control or processing logic between high level logic and the data link that maintains control of the data link. The data link level performs such functions as inserting transmit bits and deleting receive bits; interpreting address and control fields; generating, transmitting, and interpreting commands and responses; and computing and interpreting frame check sequences. See also *packet level* and *physical level*. (2) In X.25 communications, synonym for *frame level*.

**data network.** An arrangement of data circuits and switching facilities for establishing connections between data terminal equipment. (l)

**data packet.** In X.25 communications, a packet used for the transmission of user data on a virtual circuit at the DTE/DCE interface.

**data set.** (1) Synonym for *file*. (2) Deprecated term for *modem*.

**data set ready (DSR).** Synonym for *DCE ready*.

**data stream.** (1) All information (data and control commands) sent over a data link usually in a single read or write operation. (2) A continuous stream of

## Glossary

data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format.

**data terminal equipment (DTE).** That part of a data station that serves as a data source, data sink, or both. (I) (A)

**data terminal ready (DTR).** A signal to the modem used with the EIA 232 protocol.

**data types.** In the NetView program, a description of the organization of panels. Data types are alerts, events, and statistics. Data types are combined with resource types and display types to describe the NetView program's display organization. See also *display types* and *resource types*.

**DCE.** (1) Data circuit-terminating equipment. (2) Distributed Computing Environment.

**DCE clear confirmation packet.** A call supervision packet that a data circuit-terminating equipment (DCE) transmits to confirm that a call has been cleared.

**DCE ready.** In the EIA 232 standard, a signal that indicates to the data terminal equipment (DTE) that the local data circuit-terminating equipment (DCE) is connected to the communication channel and is ready to send data. Synonymous with *data set ready (DSR)*.

**ddname.** Data definition name.

**deactivate.** To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate*.

**default.** Pertaining to an attribute, condition, value, or option that is assumed when none is explicitly specified. (I)

**definite response (DR).** In SNA, a protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request chain. Contrast with *exception response* and *no response*.

**definition statement.** (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. See Figure X-1, Figure X-2, and Figure X-3. See also *macroinstruction*.

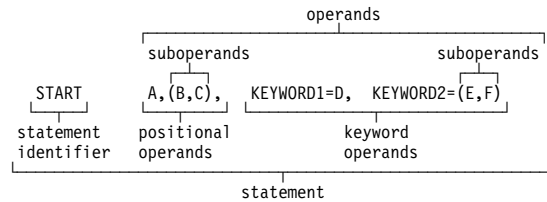


Figure X-1. Example of a Language Statement

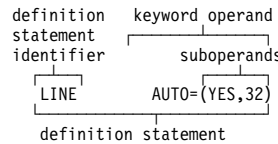


Figure X-2. Example of an NCP Definition Statement

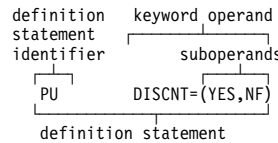


Figure X-3. Example of a VTAM Definition Statement

**degree.** In NETDA/2, the maximum number of attachments (such as links, Ethernet connections, or token-ring connections) that a node can have. For example, if a node can have six attachments, that node has a degree of 6.

**delimiter.** (1) A character used to indicate the beginning and end of a character string. (T) (2) A flag that separates and organizes items of data. (A) (3) A character that groups or separates words or values in a line of input.

**delivery-confirmation bit.** See *D-bit*.

**dependent LU.** See *SSCP-dependent LU*.

**destination.** (1) Any point or location, such as a node, station, or a particular terminal, to which information is to be sent. (2) In ACF/TCAM, the place to which a message being handled by a message handler is to be sent. (3) An external logical unit (LU) or application program to which messages or other data are directed.

**destination node.** The node to which a request or data is sent.

**device.** (1) A mechanical, electrical, or electronic contrivance with a specific purpose. (2) In the AIX operating system, a valuator, button, or the keyboard. Buttons have values of 0 or 1 (up or down); valuator return values in a range, and the keyboard returns ASCII values.

**dial-in.** Pertaining to the direction in which a switched connection is requested by any node or terminal other than the receiving host or an NCP.

**dial-out.** Pertaining to the direction in which a switched connection is requested by a host or an NCP.

**digital.** (1) Pertaining to data that consist of digits. (T) (2) Pertaining to data in the form of digits. (A) (3) Contrast with *analog*.

**direct activation.** In VTAM, the activation of a resource as a result of an activation command specifically naming the resource. See *automatic activation*. Contrast with *indirect activation*.

**directory.** (1) A table of identifiers and references to the corresponding items of data. (I) (A) (2) A database in an APPN node that lists names of resources (in particular, logical units) and records the CP name of the node where each resource is located. See *distributed directory database* and *local directory database*. (3) A type of file containing the names and controlling information for other files or other directories. (4) A listing of the files stored on a disk or diskette. (5) In VM/SP, a control program (CP) disk file that defines each virtual machine's normal configuration: the user ID, password, normal and maximum allowable virtual storage, CP command privilege classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired.

**disable.** To make nonfunctional.

**disabled.** (1) Pertaining to a state of a processing unit that prevents the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or audio response unit cannot accept incoming calls on a line. (3) In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with a disabled logical unit (LU) can specify that the session be queued by the SSCP until the LU becomes enabled. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *enabled* and *inhibited*.

**disconnection.** In VTAM, the termination of a physical connection.

**diskette.** (1) A small magnetic disk enclosed in a jacket. (T) (2) A thin, flexible magnetic disk and a semi-rigid protective jacket, in which the disk is permanently enclosed.

**display.** (1) A visual presentation of data. (I) (A) (2) To present data visually. (I) (A) (3) Deprecated term for *panel*.

**display levels.** Synonym for *display types*.

**display types.** In the NetView program, a concept to describe the organization of panels. Display types are defined as total, most recent, user action, and detail. Display types are combined with resource types and data types to describe NetView's panel organization. See also *data types* and *resource types*. Synonymous with *display levels*.

**Distributed Computing Environment (DCE).** The Open Software Foundation (OSF) specification (or a product derived from this specification) that assists in networking. DCE provides such functions as authentication, directory service (DS), and remote procedure call (RPC).

**distributed directory database.** The complete listing of all the resources in the network as maintained in the individual directories scattered throughout an APPN network. Each node has a piece of the complete directory, but it is not necessary for any one node to have the entire list. Entries are created, modified, and deleted through system definition, operator action, automatic registration, and ongoing network search procedures. Synonymous with *distributed network directory* and *network directory database*.

**distributed network directory.** Synonym for *distributed directory database*.

**DLC.** Data link control.

**DM.** Disconnect mode.

**domain.** (1) That part of a computer network in which the data processing resources are under common control. (T) (2) In SNA, see *end node domain*, *network node domain*, and *system services control point (SSCP) domain*. (3) In the Internet, a part of a naming hierarchy in which the domain name consists of a sequence of names (labels) separated by periods (dots). (4) In Open Systems Interconnection (OSI), a part of a distributed system or a set of managed objects to which a common policy applies.

**domain operator.** In a multiple-domain network, the person or program that controls operation of resources controlled by one system services control point (SSCP). See also *network operator*.

**down time.** The time during which a functional unit cannot be used because of a fault within the functional unit or within the environment. (T) (A)

**downstream.** In the direction of data flow from the host to the end user. Contrast with *upstream*.

**downstream device.** For the IBM 3710 Network Controller, a device located in a network so that the IBM 3710 is positioned between the device and a host. A

## Glossary

display terminal downstream from the 3710 is an example of a downstream device. Contrast with *upstream device*.

**DR.** (1) In VTAM, NCP, and CCP, dynamic reconfiguration. (2) In SNA, definite response.

**dsname.** Data set name.

**DSR.** Data set ready.

**DTE.** Data terminal equipment. (A)

**DTE/DCE interface.** The physical interface and link access procedures between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE).

**DTR.** Data terminal ready.

**dump.** (1) To record, at a particular instant, the contents of all or part of one storage device in another storage device. Dumping is usually for the purpose of debugging. (T) (2) Data that has been dumped. (T) (3) To copy data in a readable format from main or auxiliary storage onto an external medium such as tape, diskette, or printer. (4) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

**duplex.** Pertaining to communication in which data can be sent and received at the same time. Synonymous with *full-duplex*. Contrast with *half-duplex*.

**dynamic.** (1) In programming languages, pertaining to properties that can only be established during the execution of a program; for example, the length of a variable-length data object is dynamic. (I) (2) Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time. (3) Contrast with *static*.

**dynamic reconfiguration (DR).** The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables or deactivating the affected major node.

## E

**EBCDIC.** Extended binary-coded decimal interchange code. A coded character set of 256 8-bit characters.

**echo.** (1) In computer graphics, the immediate notification of the current values provided by an input device to the operator at the display console. (I) (A) (2) In word processing, to print or display each character or line as it is keyed in. (3) In data communication, a reflected signal on a communications channel. On a communications terminal, each signal is displayed twice, once when entered at the local terminal and

again when returned over the communications link. This allows the signals to be checked for accuracy.

**EIA.** Electronic Industries Association.

**Electronic Industries Association (EIA).** An organization of electronics manufacturers that advances the technological growth of the industry, represents the views of its members, and develops industry standards.

**element.** (1) A field in the network address. (2) In SNA, the particular resource within a subarea that is identified by an element address. See also *subarea*.

**element address.** In SNA, a value in the element address field of the network address identifying a specific resource within a subarea. See also *subarea address*.

**Emulation Program (EP).** (1) An IBM control program that allows a channel-attached IBM communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. (2) See also *network control program*.

**enable.** To make functional.

**enabled.** (1) Pertaining to a state of the processing unit that allows the occurrence of certain types of interruptions. (2) Pertaining to the state in which a transmission control unit or an audio response unit can accept incoming calls on a line. (3) In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is ready to establish LU-LU sessions. The LU can separately indicate whether this prevents it from acting as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *disabled* and *inhibited*.

**encryption.** In computer security, the process of transforming data into an unintelligible form in such a way that the original data either cannot be obtained or can be obtained only by using a decryption process.

**end bracket.** In SNA, the value (binary 1) of the end bracket indicator in the request header (RH) of the first request of the last chain of a bracket; the value denotes the end of the bracket. Contrast with *begin bracket*. See also *bracket*.

**end node (EN).** (1) See *Advanced Peer-to-Peer Networking (APPN) end node* and *low-entry networking (LEN) end node*. (2) In communications, a node that is frequently attached to a single data link and cannot perform intermediate routing functions.

**end node domain.** An end node control point, its attached links, and its local LUs.



**entry point (EP).** (1) The address or label of the first instruction executed on entering a computer program, routine, or subroutine. A computer program, routine, or subroutine may have a number of different entry points, each perhaps corresponding to a different function or purpose. (I) (A) (2) In SNA, a type 2.0, type 2.1, type 4, or type 5 node that provides distributed network management support. It sends network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated commands to manage and control its resources.

**EP.** (1) Emulation Program. (2) Entry point.

**equipment.** See *data terminal equipment (DTE)*.

**ER.** (1) Explicit route. (2) Exception response.

**exception.** An abnormal condition such as an I/O error encountered in processing a data set or a file.

**exception response (ER).** In SNA, a protocol requested in the form-of-response-requested field of a request header that directs the receiver to return a response only if the request is unacceptable as received or cannot be processed; that is, a negative response, but not a positive response, can be returned. Contrast with *definite response* and *no response*.

**exchange identification (XID).** A specific type of basic link unit that is used to convey node and link characteristics between adjacent nodes. XIDs are exchanged between link stations before and during link activation to establish and negotiate link and node characteristics, and after link activation to communicate changes in these characteristics.

**EXEC.** In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches.

**exit.** (1) To execute an instruction within a portion of a computer program in order to terminate the execution of that portion. Such portions of computer programs include loops, subroutines, modules, and so on. (T) (2) See *installation exit* and *user exit*.

**exit program.** Synonym for *exit routine*.

**exit routine.** Either of two types of routines: installation exit routines or user exit routines. Synonymous with *exit program*. See *installation exit routine* and *user exit routine*.

**explicit route (ER).** In SNA, a series of one or more transmission groups that connect two subarea nodes. An explicit route is identified by an origin subarea

address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*.

**EXT.** External trace file.

**extended architecture (XA).** An extension to System/370 architecture that takes advantage of continuing high performance enhancements to computer system hardware.

## F

**facilities field.** In an X.25 packet switching data network, an optional field used by the data terminal equipment (DTE) to convey information about the call to the network.

**fast select.** (1) An option of a virtual call facility that allows inclusion of data in call-setup and call-clearing packets. (I) (2) In X.25 communications, an optional facility that allows inclusion of data in call-request and clear-request packets.

**FASTRUN.** One of several options available with the NCP/EP definition facility (NDF) that indicates that only the syntax is to be checked in generation definition statements.

**feature.** A part of an IBM product that may be ordered separately by the customer.

**FIC.** First-in-chain.

**field.** An identifiable area in a window. Examples of fields are: an entry field, into which a user can type or place text, and a field of radio button choices, from which a user can select one choice.

**field-formatted.** Pertaining to a request or response that is encoded into fields, each having a specified format such as binary codes, bit-significant flags, and symbolic names. Contrast with *unformatted*.

**file.** A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

**first-in-chain (FIC).** A request unit (RU) whose request header (RH) begin chain indicator is on and whose RH end chain indicator is off. See also *RU chain*.

**fixed pacing.** Synonym for *fixed session-level pacing*.

**fixed session-level pacing.** A form of session-level pacing in which the data transfer rate is controlled using fixed pacing-window sizes, which are initialized at session-activation time. Synonymous with *fixed pacing*. Contrast with *adaptive session-level pacing*.

## Glossary

**flag (F) sequence.** A unique sequence of eight bits (01111110) used to delimit the opening and closing of an SDLC frame.

**flat file.** (1) A one-dimensional or two-dimensional array: a list or table of items. (2) In a relational database, synonym for *relation*. (3) A file that has no hierarchical structure.

**flow.** In NETDA/2, the amount of traffic that can pass through a node, connection, or route in both directions during a given period of time.

**flow control.** In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also  *pacing*.

**frame.** (1) In Open Systems Interconnection architecture, a data structure pertaining to a particular area of knowledge and consisting of slots that can accept the values of specific attributes and from which inferences can be drawn by appropriate procedural attachments. (T) (2) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (3) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

**frame check sequence (FCS).** A field immediately preceding the closing flag sequence of a frame that contains a bit sequence checked by the receiver to detect transmission errors.

**frame level.** Synonymous with  *data link level*. See  *link level*.

**frame reject.** In X.25, a control field that is used to indicate an error condition that cannot be recovered by the retransmission of identical frames.

**full-duplex (FDX).** Synonym for  *duplex*.

**full-screen mode.** A form of screen presentation in which the contents of an entire terminal screen can be displayed at once. Full-screen mode is often used for fill-in-the-blanks prompting.

## G

**gateway.** (1) A functional unit that interconnects two computer networks with different network architectures. A gateway connects networks or systems of different architectures. A bridge interconnects networks or systems with the same or similar architectures. (T)

(2) In the AIX operating system, an entity that operates above the link layer and translates, when required, the interface and protocol used by one network into those used by another distinct network. (3) In TCP/IP, a device used to connect two systems that use either the same or different communication protocols. (4) The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway VTAM. (5) In the IBM Token-Ring Network, a device and its associated software that connect a local area network to another local area network or a host that uses different logical link protocols. (6) Contrast with  *router*.

**general format identifier.** GFI (General Format Identifier) is the first half byte of byte 0 in the X.25 Packet Header. It identifies general information about the format of the X.25 CCITT standards.

**generation.** The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

**generation definition.** The definition statement of a resource used in generating a program.

**generic unbind.** Synonym for  *session deactivation request*.

**GID.** Group ID.

**graphic monitor.** The graphical user interface of the NetView Graphic Monitor Facility.

**group.** In the NetView/PC program, to identify a set of application programs that are to run concurrently.

**group ID (GID).** (1) In RACF, a string of one to eight characters that identifies a group. The first character must be A through Z, #, \$, or @. The rest can be A through Z, #, \$, @, or 0 through 9. (2) In the AIX operating system, a number that corresponds to a specific group name. The group ID can often be substituted in commands that take a group name as a value.

## H

**half-duplex (HD, HDX).** In data communication, pertaining to transmission in only one direction at a time. Contrast with  *duplex*.

**handle.** (1) In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and file so that the file can be found and opened. (2) In the AIX operating system, a data structure that is a temporary local identifier for an object. Allocating a handle creates it. Binding a handle

makes it identify an object at a specific location. (3) In OS/400 application programming interfaces, a variable that represents an object.

**hardcopy.** (1) A permanent copy of a display image generated on an output device such as a printer or plotter, and which can be carried away. (T) (2) A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, and summaries. (3) Contrast with *softcopy*.

**header.** (1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the message such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

**help panel.** Information displayed by a system in response to a help request from a user.

**hexadecimal.** (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (I) (2) Pertaining to a fixed-radix numeration system, with radix of 16. (I) (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

**hierarchy.** The resource types, display types, and data types that make up the organization, or levels, in a network.

**high-level data link control (HDLC).** In data communication, the use of a specified series of bits to control data links in accordance with the International Standards for HDLC: ISO 3309 Frame Structure and ISO 4335 Elements of Procedures.

**host.** (1) See *host processor*. (2) In the Internet suite of protocols, an end system. The end system can be any workstation; it does not have to be a mainframe.

**host ID.** In TCP/IP, that part of the IP address that defines the host on the network. The length of the host ID depends on the type of network or network class (A, B, or C).

**host node.** (1) A node at which a host computer is located. (T) (2) A node that provides an application programming interface (API) and a common application interface.

**host processor.** (1) A processor that controls all or part of a user application network. (T) (2) In a network, the processing unit in which the data communication access method resides.

## I

**I frame.** Information frame.

**IBM OS/2 Presentation Manager.** The front-end data manager and user interface for the IBM OS/2 operating system; an example of a graphical user interface.

**ICP.** Internet Control Protocol.

**ID.** (1) Identifier. (2) Identification.

**IMS.** Information Management System. Synonym for *IMS/VS*.

**IMS/VS.** Information Management System/Virtual Storage.

**inactive.** (1) Not operational. (2) Pertaining to a node or device not connected or not available for connection to another node or device. (3) In the AIX operating system, pertaining to a window that does not have an input focus. (4) In VTAM, the state of a resource or a major or minor node that has not been activated or for which the VARY INACT command has been issued. Contrast with *active*. See also *inoperative*.

**inbound.** In communications, data that is received from the network.

**incoming call packet.** A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a called data terminal equipment (DTE) that another DTE has requested a call.

**indirect activation.** In VTAM, the activation of a lower-level resource of the resource hierarchy as a result of SCOPE or ISTATUS specifications related to an activation command naming a higher-level resource. Contrast with *direct activation*.

**information (I) format.** A format used for information transfer.

**information (I) frame.** A frame in I format used for numbered information transfer.

**Information Management System/Virtual Storage (IMS/VS).** A database/data communication (DB/DC) system that can manage complex databases and networks. Synonymous with *IMS*.

**inhibited.** In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with an inhibited LU will be rejected by the SSCP. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *disabled* and *enabled*.

## Glossary

**initial program load (IPL).** (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs. (4) Synonymous with *system restart* and *system startup*.

**INITIATE.** A network services request sent from a logical unit (LU) to a system services control point (SSCP) requesting that an LU-LU session be established.

**INN.** Intermediate network node.

**inoperative.** The condition of a resource that has been active but is not currently active. A resource may be inoperative for reasons such as the following: a) it may have failed, b) it may have received an INOP request, or c) it may be suspended while a reactivate command is being processed. See also *inactive*.

**insert.** In LANs, to make an attaching device an active part of the LAN.

**installation.** (1) In system development, preparing and placing a functional unit in position for use. (T) (2) A particular computing system, including the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

**installation exit.** The means specifically described in an IBM software product's documentation by which an IBM software product may be modified by a customer's system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying or extending the functions of the IBM software product. Synonymous with *installation-wide exit*. See *user exit*.

**installation exit routine.** A routine written by a user to take control at an installation exit of an IBM software product.

**installation-wide exit.** Synonym for *installation exit*.

**integrated services digital network (ISDN).** A digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data.

**Note:** ISDNs are used in public and private network architectures.

**interactive.** (1) Pertaining to a program or system that alternately accepts input and then responds. An inter-

active system is conversational, that is, a continuous dialog exists between user and system. Contrast with *batch*. (2) Pertaining to the exchange of information between a user and a computer.

**interconnection.** See *SNA network interconnection (SNI)*.

**interface.** (1) A shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. (T) (2) Hardware, software, or both, that links systems, programs, or devices.

**intermediate network node (INN).** (1) In APPN, a node that is part of a route between an origin logical unit (OLU) and a destination logical unit (DLU), but does not contain the OLU or DLU and does not serve as the network server for the OLU or DLU. (2) In VTAM, deprecated term for *intermediate routing node (IRN)*. (3) In NCP, deprecated term for *subarea node (SN)*.

**intermediate routing node (IRN).** A node containing intermediate routing function.

**International Organization for Standardization (ISO).** An organization of national standards bodies from various countries established to promote development of standards to facilitate international exchange of goods and services, and develop cooperation in intellectual, scientific, technological, and economic activity.

**International Telecommunication Union (ITU).** The specialized telecommunication agency of the United Nations, established to provide standardized communication procedures and practices, including frequency allocation and radio regulations worldwide.

**Internet Control Protocol (ICP).** The Virtual NETworking System (VINES) protocol that provides exception notifications, metric notifications, and PING support. See also *RouTing update Protocol (RTP)*.

**interrupt.** (1) A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed. (A) (2) To stop a process in such a way that it can be resumed.

**IPL.** (1) Initial program loader. (A) (2) Initial program load.

**IRN.** Intermediate routing node.

**ISDN.** Integrated services digital network.

**ISO.** International Organization for Standardization.

**ISTATUS.** In VTAM and NCP, a definition specification method for indicating the initial status of resources. See also *indirect activation*.

**item.** In CCP, any of the components, such as communication controllers, lines, cluster controllers, and terminals, that comprise an IBM 3710 Network Controller configuration.

**ITU.** International Telecommunication Union.

**ITU-T.** See *ITU-TS*.

**ITU-TS.** International Telecommunication Union - Telecommunication Standardization Sector. The part of the International Telecommunication Union (ITU) that is responsible for developing recommendations for telecommunications.

## J

**JCL.** Job control language.

**job control language (JCL).** A control language used to identify a job to an operating system and to describe the job's requirements.

## K

**Kbps.** Kilobits per second.

**keyword.** (1) In programming languages, a lexical unit that, in certain contexts, characterizes some language construct; for example, in some contexts, IF characterizes an if-statement. A keyword normally has the form of an identifier. (I) (2) One of the predefined words of an artificial language. (A) (3) A significant and informative word in a title or document that describes the content of that document. (4) A name or symbol that identifies a parameter. (5) The part of a command operand that consists of a specific character string (such as DSNAME=). See also *definition statement* and *keyword operand*. Contrast with *positional operand*.

**keyword operand.** An operand that consists of a keyword followed by one or more values (such as DSNAME=HELLO). See also *definition statement*. Contrast with *positional operand*.

**keyword parameter.** A parameter that consists of a keyword followed by one or more values.

## L

**LAP.** Link access procedures.

**LAPB.** Link access protocol balanced.

**last-in-chain (LIC).** A request unit (RU) whose request header (RH) end chain indicator is on and whose RH begin chain indicator is off. See also *RU chain*.

**layer.** (1) In network architecture, a group of services that is complete from a conceptual point of view, that is one out of a set of hierarchically arranged groups, and that extends across all systems that conform to the network architecture. (T) (2) In the Open Systems Interconnection reference model, one of seven conceptually complete, hierarchically arranged groups of services, functions, and protocols, that extend across all open systems. (T) (3) In SNA, a grouping of related functions that are logically separate from the functions in other groups. Implementation of the functions in one layer can be changed without affecting functions in other layers.

**leased line.** Synonym for *nonswitched line*.

**LEN.** Low-entry networking.

**LIC.** (1) Last-in-chain. (2) In NCP, line interface coupler.

**line.** (1) The portion of a data circuit external to data circuit-terminating equipment (DCE), that connects the DCE to a data switching exchange (DSE), that connects a DCE to one or more other DCEs, or that connects a DSE to another DSE. (I) (2) Synonymous with *channel* and *circuit*.

**line control discipline.** Synonym for *link protocol and protocol*.

**line discipline.** Synonym for *link protocol and protocol*.

**line group.** One or more telecommunication lines of the same type that can be activated and deactivated as a unit.

**line mode.** A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with *full-screen mode*.

**line speed.** The number of binary digits that can be sent over a telecommunication line in one second, expressed in bits per second (bps).

**line switching.** Synonym for *circuit switching*.

## Glossary

**link.** (1) The combination of the link connection (the transmission medium) and two link stations, one at each end of the link connection. A link connection can be shared among multiple links in a multipoint or token-ring configuration. (2) To interconnect items of data or portions of one or more computer programs: for example, the linking of object programs by a linkage editor, linking of data items by pointers. (T) (3) In SNA, synonymous with *data link*.

**link access procedures (LAP).** The link level elements used for data interchange between data circuit-terminating equipment (DCE) and data terminal equipment (DTE) operating in user classes of service 8 to 11, as specified in CCITT Recommendation X.1.

**link access protocol balanced (LAPB).** A protocol used for accessing an X.25 network at the link level. LAPB is a duplex, asynchronous, symmetric protocol, used in point-to-point communication.

**link-attached.** Pertaining to devices that are connected to a controlling unit by a data link. Contrast with *channel-attached*. Synonymous with *remote*.

**link connection.** (1) The physical equipment providing two-way communication between one link station and one or more other link stations; for example, a telecommunication line and data circuit-terminating equipment (DCE). (2) In SNA, synonymous with *data circuit*.

**link connection segment.** A portion of the configuration that is located between two resources listed consecutively in the service point command service (SPCS) query link configuration request list.

**link level.** A part of Recommendation X.25 that defines the link protocol used to get data into and out of the network across the full-duplex link connecting the subscriber's machine to the network node. LAP and LAPB are the link access protocols recommended by the CCITT. See *data link level*.

**Link Problem Determination Aid (LPDA).** A series of procedures that are used to test the status of and to control DCEs, the communication line, and the remote device interface. These procedures, or a subset of them, are implemented by host programs (such as the NetView program and VTAM), communication controller programs (such as NCP), and IBM LPDA DCEs. See also *LPDA-1* and *LPDA-2*.

**link protocol.** (1) The rules for sending and receiving data at the link level. (2) Synonymous with *line control discipline* and *line discipline*.

**link station.** (1) The hardware and software components within a node representing a connection to an adjacent node over a specific link. For example, if node A is the primary end of a multipoint line that connects to three adjacent nodes, node A will have three link

stations representing the connections to the adjacent nodes. See also *adjacent link station (ALS)*. (2) In VTAM, a named resource within an APPN or a subarea node that represents the connection to another APPN or subarea node that is attached by an APPN or a subarea link. In the resource hierarchy in a subarea network, the link station is subordinate to the subarea link.

**LLC.** Logical link control.

**LL2.** Link level 2.

**load.** (1) To bring all or part of a computer program into memory from auxiliary storage so that the computer can run the program. (2) To place a diskette into a diskette drive.

**load module.** All or part of a computer program in a form suitable for loading into main storage for execution. A load module is usually the output of a linkage editor. (T)

**local.** Pertaining to a device accessed directly without use of a telecommunication line. Contrast with *remote*. Synonym for *channel-attached*.

**local address.** In SNA, an address used in a peripheral node in place of a network address and transformed to or from a network address by the boundary function in a subarea node.

**local directory database.** That set of resources (LUs) in the network known at a particular node. The resources included are all those in the node's domain as well as any cache entries.

**local host.** In TCP/IP, the host on the network at which a particular operator is working.

**Locate.** Synonym for *Locate/CD-Initiate*.

**Locate search message.** Synonym for *Locate/CD-Initiate*.

**Locate/CD-Initiate.** (1) An abbreviated term for a message exchanged between APPN nodes that contains one of the following sets of general data stream (GDS) variables:

- A Locate, a Find Resource, and a Cross-Domain Initiate GDS variable used for a network search request
- A Locate, a Found Resource, and a Cross-Domain Initiate GDS variable used for a search reply when a network resource has been located

These message structures correspond to the CP components that perform the search of the distributed network directory and establish the session. The Locate GDS variable contains information used to

control the delivery of the search messages in the network. The Find and Found GDS variables contain information used in the directories: origin cache data (control point information) and search arguments (destination LU name), and located resource information, respectively. The Cross-Domain Initiate GDS variable contains endpoint TG vector information to be used in selecting the route for the session. The length of the Locate/CD-Initiate message is limited to 1024 bytes. (2) Synonymous with *Locate* and *Locate search message*.

**logged-on operator.** A NetView operator station task that requires a terminal and a logged-on user. Contrast with *autotask*.

**logical channel.** In packet mode operation, a sending channel and a receiving channel that together are used to send and receive data over a data link at the same time. Several logical channels can be established on the same data link by interleaving the transmission of packets.

**logical channel identifier.** A bit string in the header of a packet that associates the packet with a specific switched virtual circuit or permanent virtual circuit.

**logical line.** In NCP, the representation of the connection between NCP and a node communicating with NCP over a physical line such as token-ring or frame-relay. A single physical line can support multiple logical lines. Contrast with *physical line*.

**logical link.** (1) A pair of link stations, one in each of two adjacent nodes, and their underlying link connection, providing a single link-layer connection between the two nodes. Multiple logical links can be distinguished while they share the use of the same physical media connecting two nodes. Examples are 802.2 logical links used on local area network (LAN) facilities and LAP E logical links on the same point-to-point physical link between two nodes. The term logical link also includes the multiple X.25 logical channels that share the use of the access link from a DTE to an X.25 network. (2) In APPNTAM, the unidirectional representation in a node of a link.

**logical link control (LLC).** The data link control (DLC) LAN sublayer that provides two types of DLC operation for the orderly exchange of information. The first type is connectionless service, which allows information to be sent and received without establishing a link. The LLC sublayer does not perform error recovery or flow control for connectionless service. The second type is connection-oriented service, which requires establishing a link prior to the exchange of information. Connection-oriented service provides sequenced information transfer, flow control, and error recovery.

**logical link control (LLC) protocol.** In a local area network, the protocol that governs the exchange of transmission frames between data stations independently of how the transmission medium is shared. (T) The LLC protocol was developed by the IEEE 802 committee and is common to all LAN standards.

**logical link control (LLC) protocol data unit.** A unit of information exchanged between link stations in different nodes. The LLC protocol data unit contains a destination service access point (DSAP), a source service access point (SSAP), a control field, and user data.

**logical unit (LU).** A type of network accessible unit that enables end users to gain access to network resources and communicate with each other.

**logmode table.** Synonym for *logon mode table*.

**logon.** (1) The procedure by which a user begins a terminal session. (2) In VTAM, an unformatted session-initiation request for a session between two logical units.

**logon data.** In VTAM, either of the following:

- The user data portion of a field-formatted or unformatted session-initiation request
- The entire logon sequence or message from a logical unit (LU).

Synonymous with *logon message*.

**logon message.** Synonym for *logon data*.

**logon mode.** In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters*.

**logon mode table.** In VTAM, a set of entries for one or more logon modes. Each logon mode is identified by a logon mode name. Synonymous with *logmode table*.

**low-entry networking (LEN).** A capability of nodes to attach directly to one another using basic peer-to-peer protocols to support multiple and parallel sessions between logical units.

**low-entry networking (LEN) end node.** A LEN node receiving network services from an adjacent APPN network node.

**low-entry networking (LEN) node.** A node that provides a range of end-user services, attaches directly to other nodes using peer protocols, and derives network services implicitly from an adjacent APPN network node, that is, without the direct use of CP-CP sessions.

**LPDA.** Link Problem Determination Aid.

## Glossary

**LPDA-1.** The first version of the LPDA command set. LPDA-1 is not compatible with LPDA-2. See also *Link Problem Determination Aid (LPDA)* and *LPDA-2*.

**LPDA-2.** The second version of the LPDA command set. LPDA-2 provides all of the functions of LPDA-1; it also supports commands such as the following:

- DCE configuration
- Dial
- Set transmit speed
- Commands to operate a contact that can control external devices.

See also *Link Problem Determination Aid (LPDA)* and *LPDA-1*.

**LU.** Logical unit.

**LU-LU session.** A logical connection between two logical units (LUs) in an SNA network that typically provides communication between two end users.

**LU type.** The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:

- The mandatory and optional values allowed in the session activation request
- The usage of data stream controls, function management headers (FMHs), request unit parameters, and sense data values
- Presentation services protocols such as those associated with FMH usage

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

**LU 6.2.** A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application programming interface (API) consisting of structured verbs that are mapped into a product implementation.

**LUCB.** Logical unit control block.

**LUS.** Logical unit services.

## M

**M-bit.** More-data bit. In X.25 communications, the bit in a data packet that indicates that there is more data to follow in another data packet, when a message is too large for one packet.

**MAC.** Medium access control.

**machine check handler (MCH).** A feature that analyzes errors and attempts recovery by retrying the failing instruction. If retry is unsuccessful, it attempts to correct the malfunction or to isolate the affected task.

**macroinstruction.** (1) An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language and that may also specify values for parameters in the replaced instructions. (T) (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program. See also *definition statement*.

**maintain system history program (MSHP).** A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

**maintenance analysis procedure (MAP).** A maintenance document that gives an IBM service representative a step-by-step procedure for tracing a symptom to the cause of a failure.

**maintenance services.** In SNA, one of the types of network services in system services control points (SSCPs) and physical units (PUs). Maintenance services provide facilities for testing links and nodes and for collecting and recording error information.

**major node.** In VTAM, a set of resources that can be activated and deactivated as a group. See *minor node*.

**management services (MS).** (1) One of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management, and change management. (2) Services that assist in the management of systems and networks in areas such as problem management, performance management, business management, operations management, configuration management, and change management.

**manager.** (1) In OSI management, a user that, for a particular interaction, has assumed a manager role. (2) A system that assumes a manager role.

**MAP.** Maintenance analysis procedure.

**mapping.** The process of converting data that is transmitted in one format by the sender into the data format that can be accepted by the receiver.

**matching.** In MPTN architecture, pertaining to the relationship between peer transport users or peer transport providers that use the same user protocols or the same transport protocols.



**MCH.** Machine check handler.

**medium access control (MAC).** In LANs, the sublayer of the data link control layer that supports medium-dependent functions and uses the services of the physical layer to provide services to the logical link control (LLC) sublayer. The MAC sublayer includes the method of determining when a device has access to the transmission medium.

**medium access control (MAC) protocol.** In a local area network, the protocol that governs access to the transmission medium, taking into account the topological aspects of the network, in order to enable the exchange of data between data stations. (T)

**medium access control (MAC) sublayer.** In a local area network, the part of the data link layer that applies a medium access method. The MAC sublayer supports topology-dependent functions and uses the services of the physical layer to provide services to the logical link control sublayer. (T)

**megabyte (MB).** (1) For processor storage and real and virtual memory,  $2^{20}$  or 1 048 576 bytes. (2) For disk storage capacity and transmission rates, 1 000 000 bytes.

**message.** (1) An assembly of characters and sometimes control codes that is transferred as an entity from an originator to one or more recipients. A message consists of two parts: envelope and content. (T) (2) In VTAM, the amount of function management data (FMD) transferred to VTAM by the application program with one SEND request.

**message switching.** The process of receiving a message, storing it, and forwarding it to its destination unaltered. (T)

**method.** In the NetView program, the code that runs within the Resource Object Data Manager (RODM) address space. Methods are used to implement behavior specified by an operation.

**metric.** In Internet communications, a value, associated with a route, which is used to discriminate between multiple exit or entry points to the same autonomous system. The route with the lowest metric is preferred.

**MIC.** Middle-in-chain.

**microcode.** (1) One or more microinstructions. (2) A code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program-addressable. (3) To design, write, and test one or more microinstructions.

**middle-in-chain (MIC).** A request unit (RU) whose request header (RH) begin chain indicator and RH end chain indicator are both off. See also *RU chain*.

**migration.** The installation of a new version or release of a program to replace an earlier version or release.

**minor node.** In VTAM, a uniquely defined resource within a major node. See *major node* and *node*.

**mixed-media multilink transmission group (MMMLTG).** See *transmission group (TG)*.

**MLTG.** Multilink transmission group.

**MMMLTG.** Mixed-media multilink transmission group.

**mode.** See *mode name*.

**mode name.** The name used by the initiator of a session to designate the characteristics desired for the session, such as traffic pacing values, message-length limits, sync point and cryptography options, and the class of service within the transport network.

**modem (modulator/demodulator).** (1) A functional unit that modulates and demodulates signals. One of the functions of a modem is to enable digital data to be transmitted over analog transmission facilities. (T) (A) (2) A device that converts digital data from a computer to an analog signal that can be transmitted on a telecommunication line, and converts the analog signal received to data for the computer.

**module.** A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine. (A)

**modulo.** (1) Pertaining to a modulus; for example, 9 is equivalent to 4 modulo 5. (2) See also *modulus*.

**modulus.** A number, such as a positive integer, in a relationship that divides the difference between two related numbers without leaving a remainder; for example, 9 and 4 have a modulus of 5 ( $9 - 4 = 5$ ;  $4 - 9 = -5$ ; and 5 divides both 5 and -5 without leaving a remainder).

**monitor.** (1) A device that observes and records selected activities within a data processing system for analysis. Possible uses are to indicate significant departure from the norm, or to determine levels of utilization of particular functional units. (T) (2) Software or hardware that observes, supervises, controls, or verifies operations of a system. (A) (3) The function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations. (4) In the NetView Graphic Monitor Facility, to open a view that can receive status changes from the NetView program. Problem determi-

## Glossary

nation and correction can be performed directly from the view. Contrast with *browse*.

**more-data bit.** See *M-bit*.

**MSG.** Console messages.

**MSHP.** Maintain system history program.

**multidrop line.** Synonym for *multipoint line*.

**multilink transmission group (MLTG).** See *transmission group (TG)*.

**Multiple Virtual Storage (MVS).** See *MVS*.

**multipoint line.** A telecommunication line or circuit that connects two or more stations. Synonymous with *multidrop line*. Contrast with *point-to-point line*.

**MVS.** Multiple Virtual Storage. Implies MVS/370, the MVS/XA product, and the MVS/ESA product.

## N

**native.** In MPTN architecture, pertaining to the relationship between a transport user and a transport provider that are both based on the same transport protocol.

**native network.** (1) The subnetwork whose network identifier a node uses for its own network-qualified resource names. (2) In MPTN architecture, with respect to a particular transport user, a transport network that provides the address type and transport characteristics assumed in the design of the transport user. No MPTN address mapping or compensation protocols are used for data transfer.

**NCP.** (1) Network Control Program. (2) Synonymous with *ACF/NCP*.

**NCP/EP definition facility (NDF).** A program that is part of System Support Programs (SSP) and that is used to generate a load module for a partitioned emulation program (PEP), a Network Control Program (NCP), or an Emulation Program (EP).

**NDF.** NCP/EP definition facility.

**negative response (NR).** In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*.

**negotiation.** The process of deciding what packet size to transmit between a network and a 3710 Network Controller.

**net ID.** See *network identifier*.

**NETID.** See *network identifier*.

**NetView Bridge.** A set of application programming interfaces that allow the NetView program to interact with various types of databases in the MVS environment.

**NetView Graphic Monitor Facility (NGMF).** A function of the NetView program that provides the network operator with a graphic topological presentation of a network controlled by the NetView program and that allows the operator to manage the network interactively.

**NetView Installation and Administration Facility/2 (NIAF/2).** An OS/2-based tool that allows new NetView users or users migrating from a prior release of NetView to install, administer, and maintain the NetView program. NIAF/2 replaces the Interactive System Productivity Facility-based (ISPF-based) NetView Installation Facility.

**NetView-NetView task (NNT).** The task under which a cross-domain NetView operator session runs. See *operator station task*.

**NetView Performance Monitor (NPM).** An IBM licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

**NetView program.** An IBM licensed program used to monitor and manage a network and to diagnose network problems.

**network.** (1) An arrangement of nodes and connecting branches. (T) (2) A configuration of data processing devices and software connected for information interchange. (3) A group of nodes and the links interconnecting them.

**network address.** (1) In a subarea network, an address, consisting of subarea and element fields, that identifies a link, link station, physical unit, logical unit, or system services control point. Subarea nodes use network addresses; peripheral nodes use local addresses or local-form session identifiers (LFSIDs). The boundary function in the subarea node to which a peripheral node is attached transforms local addresses or LFSIDs to network addresses and vice versa. Contrast with *network name*. (2) According to ISO 7498-3, a name, unambiguous within the OSI environment, that identifies a set of network service access points.

**network architecture.** The logical structure and operating principles of a computer network. (T)

**Note:** The operating principles of a network include those of services, functions, and protocols.

**network control (NC).** In SNA, a request/response unit (RU) category used for requests and responses exchanged between physical units (PUs) for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjust peripheral nodes. See also *data flow control*, *function management data*, and *session control*.

**network control program.** A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

**Network Control Program (NCP).** An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

**network controller.** A concentrator and protocol converter used with SDLC links. By converting protocols, which manage the way data is sent and received, the IBM 3710 Network Controller allows the use of non-SNA devices with an SNA host processor.

**network directory database.** Synonym for *distributed directory database*.

**network identifier.** (1) In TCP/IP, that part of the IP address that defines a network. The length of the network ID depends on the type of network class (A, B, or C). (2) A 1- to 8-byte customer-selected name or an 8-byte IBM-registered name that uniquely identifies a specific subnetwork. (3) In MPTN architecture, the address qualifier of a transport provider address that identifies a group of nodes according to the network in which they reside.

**network name.** (1) The symbolic identifier by which end users refer to a network accessible unit, a link, or a link station within a given subnetwork. In APPN networks, network names are also used for routing purposes. Contrast with *network address*. (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program. The network name must be unique across domains. See *uninterpreted name*.

**network node (NN).** See *Advanced Peer-to-Peer Networking (APPN) network node*.

**network node domain.** An APPN network-node control point, its attached links, the network resources for which it answers directory search requests (namely, its local LUs and adjacent LEN end nodes), the adjacent APPN end nodes with which it exchanges directory search requests and replies, and other resources (such as a local storage device) associated with its own node or an adjacent end node for which it provides management services.

**network operator.** (1) A person who controls the operation of all or part of a network. (2) In a multiple-

domain network, a person or program responsible for controlling all domains. Contrast with *domain operator*.

**network performance analyzer (NPA).** A function of NCP that collects performance data about devices. The data is recorded by NPM.

**Network Routing Facility (NRF).** An IBM licensed program that resides in NCP. NRF provides a path for routing messages between terminals and routes messages over this path without going through the host processor.

**network services.** (1) The services within network accessible units that control network operation through SSCP-SSCP, SSCP-PU, SSCP-LU, and CP-CP sessions. (2) The session services (directory and route-selection functions) and management services provided by an APPN network-node control point to its domain.

**Network Terminal Option (NTO).** An IBM licensed program, used in conjunction with NCP, that allows certain non-SNA devices to participate in sessions with SNA application programs in the host processor. When data is sent from a non-SNA device to the host processor, NTO converts non-SNA protocol to SNA protocol; and when data is sent from the host processor to the non-SNA device, NTO converts SNA protocol to non-SNA protocol.

**network user address (NUA).** In X.25 communications, the X.121 address containing up to 15 binary code digits.

**NIAF/2.** NetView Installation and Administration Facility/2.

**NN.** Network node.

**no response.** In SNA, a protocol requested in the form-of-response-requested field of the request header that directs the receiver of the request not to return any response, regardless of whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

**node.** (1) In a network, a point at which one or more functional units connect channels or data circuits. (1) (2) Any device, attached to a network, that transmits and receives data. (3) An endpoint of a link or a junction common to two or more links in a network. Nodes can be processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. (4) In VTAM, a point in a network defined by a symbolic name. See *major node* and *minor node*.

**node type.** A designation of a node according to the protocols it supports or the role it plays in a network. Node type was originally denoted numerically (as 1, 2.0, 2.1, 4, and 5) but is now characterized more specifically

## Glossary

by protocol type (APPN network node, LEN node, subarea node, and interchange node, for example) because type 2.1 nodes and type 5 nodes support multiple protocol types and roles.

**nonswitched connection.** A connection that does not have to be established by dialing. Contrast with *switched connection*.

**nonswitched line.** A telecommunication line on which connections do not have to be established by dialing. Synonymous with *leased line*. Contrast with *switched line*.

**notification.** (1) An unscheduled, spontaneously generated report of an event that has occurred. (2) In OSI management, information emitted by a managed object relating to an event that has occurred within the managed object.

**NOTIFY.** A network services request that is sent by a system services control point (SSCP) to a logical unit (LU) to inform the LU of the status of a procedure requested by the LU.

**NPA.** Network performance analyzer.

**NPDA.** (1) Network Problem Determination Application. (2) A command that starts the NetView hardware monitor. NPDA also identifies various panels and functions as part of the hardware monitor.

**NPM.** NetView Performance Monitor.

**NPSI.** X.25 NCP Packet Switching Interface.

**NRF.** Network Routing Facility.

**NRZ-1.** Non-return-to-zero change-on-ones recording. (I) (A)

**NRZI.** Non-return-to-zero (inverted) recording. Deprecated term for *non-return-to-zero change-on-ones recording (NRZ-1)*.

**NTO.** Network Terminal Option.

## O

**object.** (1) In object-oriented design or programming, an abstraction consisting of data and the operations associated with that data. See also *class*. (2) An item that a user can manipulate as a single unit to perform a task. An object can appear as text, an icon, or both.

**ODLC.** Outboard data link control.

**offline.** (1) Pertaining to the operation of a functional unit that takes place either independently of, or in parallel with, the main operation of a computer. (T)

(2) Neither controlled by, nor communicating with, a computer. Contrast with *online*.

**offset.** The number of measuring units from an arbitrary starting point in a record, area, or control block, to some other point.

**OIC.** Only-in-chain.

**online.** (1) Pertaining to the operation of a functional unit when under the direct control of the computer. (T) (2) Pertaining to a user's ability to interact with a computer. (A) (3) Pertaining to a user's access to a computer via a terminal. (A) (4) Controlled by, or communicating with, a computer. (5) Contrast with *offline*.

**only-in-chain (OIC).** A request unit for which the request header (RH) begin chain indicator and RH end chain indicator are both on. See also *RU chain*.

**open.** (1) A break in an electrical circuit. (2) To make an adapter ready for use.

**open system.** A system whose characteristics comply with standards made available throughout the industry and that therefore can be connected to other systems complying with the same standards. (T)

**Open Systems Interconnection (OSI).** (1) The interconnection of open systems in accordance with standards of the International Organization for Standardization (ISO) for the exchange of information. (T) (A) (2) The use of standardized procedures to enable the interconnection of data processing systems.

**Note:** OSI architecture establishes a framework for coordinating the development of current and future standards for the interconnection of computer systems. Network functions are divided into seven layers. Each layer represents a group of related data processing and communication functions that can be carried out in a standard way to support different applications.

**Open Systems Interconnection (OSI) architecture.** Network architecture that adheres to that particular set of ISO standards that relates to Open Systems Interconnection. (T)

**Open Systems Interconnection (OSI) reference model.** A model that describes the general principles of the Open Systems Interconnection, as well as the purpose and the hierarchical arrangement of its seven layers. (T)

**operand.** (1) An entity on which an operation is performed. (I) (2) That which is operated upon. An operand is usually identified by an address part of an instruction. (A) (3) Information entered with a command

name to define the data on which a command processor operates and to control the execution of the command processor. (4) An expression to whose value an operator is applied. See also *definition statement*, *keyword*, *keyword parameter*, and *parameter*.

**operating system (OS).** Software that controls the execution of programs and that may provide services such as resource allocation, scheduling, input/output control, and data management. Although operating systems are predominantly software, partial hardware implementations are possible. (T)

**operation.** In object-oriented design or programming, a service that can be requested at the boundary of an object. Operations include modifying an object or disclosing information about an object.

**operator.** (1) In a language statement, the lexical entity that indicates the action to be performed on operands. See also *definition statement*. (2) A person or program responsible for managing activities controlled by a given piece of software such as MVS, the NetView program, or IMS. See *logged-on operator* and *network operator*. See also *autotask* and *operator station task*. (3) A person who operates a device. (4) A person who keeps a system running.

**operator station task (OST).** The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program. See *NetView-NetView task*.

**OPNDST.** Open destination.

**origin.** An external logical unit (LU) or application program from which a message or other data originates. See also *destination*.

**OS.** Operating system.

**OSI.** Open Systems Interconnection.

**outboard data link control (ODLC).** (1) Data link control (DLC) processing performed by a coprocessor. (2) In NCP, data link control (DLC) processing performed by the 3746 Model 900 Connectivity Subsystem (CSS). (3) For the IBM 6611 Network Processor, data link control (DLC) processing performed by a deep adapter.

**outbound.** In communications, data that is transmitted to the network.

## P

**pacing.** A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See also *flow control*, *receive pacing*, *send pacing*, *session-level pacing*, and *virtual route (VR) pacing*.

**pacing group.** Synonym for *pacing window*.

**pacing response.** In SNA, an indicator that signifies the readiness of a receiving component to accept another pacing group. The indicator is carried in a response header (RH) for session-level pacing and in a transmission header (TH) for virtual route pacing.

**pacing window.** (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual route receiver is ready to receive more PIUs on the route. (2) The requests that can be transmitted on the normal flow in one direction on a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests. (3) Synonymous with *pacing group*.

**packet.** In data communication, a sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and, possibly, error control information are arranged in a specific format. (I)

**packet assembler/disassembler (PAD).** A functional unit that enables data terminal equipment (DTEs) not equipped for packet switching to access a packet switched network. (T) (A)

**packet level.** (1) The packet format and control procedures for exchange of packets containing control information and user data between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). See also *data link level* and *physical level*. (2) A part of Recommendation X.25 that defines the protocol for establishing logical connections between two DTEs and for transferring data on these connections.

**packet mode operation.** Synonym for *packet switching*.

**packet modulo.** The highest sequence number the packet level uses before resetting the count and beginning the count again.

**packet sequencing.** A process of ensuring that packets are delivered to the receiving data terminal equipment (DTE) in the same sequence in which they were transmitted by the sending DTE. (I) (A)

## Glossary

**packet size.** (1) In X.25 communications, the length of the user data in a data packet. (2) The maximum amount of user data in a packet.

**packet switching.** (1) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during transmission of a packet. On completion of the transmission, the channel is made available for transfer of other packets. (I) (2) Synonymous with *packet mode operation*. See also *circuit switching*.

**packet switching data network (PSDN).** A network that uses packet switching as a means of transmitting data.

**packet window.** The maximum number of consecutive data packets that are allowed to flow between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) before an acknowledgment is received for a given logical channel.

**PAD.** Packet assembler/disassembler.

**page.** (1) In a virtual storage system, a fixed-length block that has a virtual address and is transferred as a unit between real storage and auxiliary storage. (I) (A) (2) A printed form. (3) The information displayed at the same time on the screen of a display device. (4) In VSE, a fixed-length block of instructions, data, or both that can be located in processor storage or in the page data set on disk. (5) To replace the information displayed on the screen with prior or subsequent information from the same file.

**panel.** (1) See *window*. (2) A formatted display of information that appears on a display screen. See *help panel* and *task panel*. (3) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

**parallel.** (1) Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus. (T) (2) Pertaining to concurrent or simultaneous operation of two or more devices or to concurrent performance of two or more activities in a single device. (A) (3) Pertaining to concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels. (A) (4) Pertaining to the simultaneity of two or more processes. (A) (5) Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts. (A) (6) Contrast with *serial*.

**parallel sessions.** Two or more concurrently active sessions between the same two network accessible

units (NAUs) using different pairs of network addresses or local-form session identifiers. Each session can have independent session parameters.

**parallel transmission groups.** Multiple transmission groups between adjacent nodes, with each group having a distinct transmission group number.

**parameter.** (1) A variable that is given a constant value for a specified application and that may denote the application. (I) (A) (2) In Basic CUA architecture, a variable used in conjunction with a command to affect its result. (3) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (4) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. See also *keyword*, *keyword parameter*, and *operand*.

**parameters.** In NETDA/2, the set of restrictions that affect only the output of a network design. A change in a parameter value does not change the input to the network design. Contrast with *constraints*.

**partitioned data set (PDS).** A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**partitioned emulation programming (PEP) extension.** A function of a network control program that enables a communication controller to operate some telecommunication lines in network control mode while simultaneously operating others in emulation mode.

**password.** (1) A value used in authentication or a value used to establish membership in a set of people having specific privileges. (2) A unique string of characters known to a computer system and to a user, who must specify the character string to gain access to a system and to the information stored within it. (3) In computer security, a string of characters known to the computer system and a user, who must specify it to gain full or limited access to a system and to the data stored within it.

**path.** (1) In a network, any route between any two nodes. A path may include more than one branch. (T) (2) The series of transport network components (path control and data link control) that are traversed by the information exchanged between two network accessible units. See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*. (3) In VTAM when defining a switched major node, a potential dial-out port that can be used to reach that node. (4) In the NetView/PC program, a complete line in a configuration that contains

all of the resources in the service point command service (SPCS) query link configuration request list.

**path information unit (PIU).** A message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment.

**path statement.** In NETDA/2, a type of route statement used for subarea routing. Existing VTAM and NCP path statements can be included in a network design, or new path statements can be generated.

**PDS.** Partitioned data set.

**peer.** In network architecture, any functional unit that is in the same layer as another entity. (T)

**pending active session.** In VTAM, the state of an LU-LU session recorded by the system services control point (SSCP) when it finds both logical units (LUs) available and has sent a CINIT request to the primary logical unit (PLU) of the requested session.

**PEP.** Partitioned emulation programming.

**peripheral link.** In SNA, a link between a subarea and a peripheral node. See also *route extension (REX)*.

**peripheral node.** A node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node can be a type 1, 2.0, or 2.1 node connected to a subarea boundary node.

**peripheral PU.** In SNA, a physical unit in a peripheral node. Contrast with *subarea PU*.

**permanent virtual circuit (PVC).** (1) In X.25 and frame-relay communications, a virtual circuit that has a logical channel permanently assigned to it at each data terminal equipment (DTE). Call-establishment protocols are not required. Contrast with *switched virtual circuit (SVC)*. (2) The logical connection between two frame-relay terminating equipment stations, either directly or through one or more frame-relay frame handlers. A PVC consists of one or more PVC segments.

**physical circuit.** A circuit established without multiplexing. See also *data circuit*. Contrast with *virtual circuit*.

**physical connection.** (1) A connection that establishes an electrical circuit. (2) In VTAM, a point-to-point or multipoint connection.

**physical level.** In X.25, the mechanical, electrical, functional, and procedural media used to activate, maintain, and deactivate the physical link between the data terminal equipment (DTE) and the data circuit-

terminating equipment (DCE). See *data link level* and *packet level*.

**physical line.** In NCP, the physical connection between NCP and an adjacent device or local area network (LAN). A single physical line, such as token-ring or frame-relay, can support multiple logical lines. Contrast with *logical line*.

**physical services header (PSH).** The protocol above the X.25 packet level used to provide adjacent node services for SNA-to-SNA communication to IBM 5973 Network Interface Adapters.

**physical unit (PU).** The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. See also *peripheral PU* and *subarea PU*.

**physical unit (PU) services.** In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions.

**piggybacking.** When data is sent between two NCPs or between an NCP and a link-attached station, the process of combining a PIU and an acknowledgment for a PIU that was previously sent.

**PIU.** Path information unit.

**PLU.** Primary logical unit.

**PM.** Presentation Manager.

**point-to-point line.** A switched or nonswitched telecommunication line that connects a single remote station to a computer. Contrast with *multipoint line*.

**Point-to-Point Protocol (PPP).** A protocol that provides a method for encapsulating and transmitting packets over serial point-to-point links.

**polling.** (1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (1) (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (A)

**port.** (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and printers are attached. Synonymous with *socket*. (3) The representation of a physical connection to the link hardware. A port is sometimes referred to as an adapter; however, there

## Glossary

can be more than one port on an adapter. There may be one or more ports controlled by a single DLC process. (4) In the Internet suite of protocols, a 16-bit number used to communicate between TCP or the User Datagram Protocol (UDP) and a higher-level protocol or application. Some protocols, such as File Transfer Protocol (FTP) and Simple Mail Transfer Protocol (SMTP), use the same well-known port number in all TCP/IP implementations. (5) An abstraction used by transport protocols to distinguish among multiple destinations within a host machine.

**positional operand.** An operand in a language statement that has a fixed position. See also *definition statement*. Contrast with *keyword operand*.

**positive response.** In SNA, a response indicating that a request was received and processed. Contrast with *negative response*.

**PP.** Parallel print.

**PPP.** Point-to-Point Protocol.

**preferential closed user group (CUG).** In X.25 communications, the default closed user group.

**Presentation Manager (PM).** See *IBM OS/2 Presentation Manager*.

**primary logical unit (PLU).** In SNA, the logical unit (LU) that sends the BIND to activate a session with its partner LU. Contrast with *secondary logical unit (SLU)*.

**problem determination.** The process of determining the source of a problem; for example, a program component, machine failure, telecommunication facilities, user or contractor-installed programs or equipment, environmental failure such as a power loss, or user error.

**process.** (1) A course of the events defined by its purpose or by its effect, achieved under given conditions. (2) In data processing, the course of events that occurs during the execution of all or part of a program. (T) (3) Any operation or combination of operations on data. (4) A function being performed or waiting to be performed. (5) To perform operations on data in a process. (I) (A)

**process identification number (process ID).** A unique number assigned to a process by the operating system. The number is used internally by processes to communicate.

**processor.** In a computer, a functional unit that interprets and executes instructions. A processor consists of at least an instruction control unit and an arithmetic and logic unit. (T)

**profile.** Data that describes the significant characteristics of a user, a group of users, or one or more computer resources.

**program operator.** A VTAM application program that is authorized to issue VTAM operator commands and receive VTAM operator awareness messages. See also *solicited message* and *unsolicited message*.

**program temporary fix (PTF).** A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

**protocol.** (1) A set of semantic and syntactic rules that determine the behavior of functional units in achieving communication. (I) (2) In Open Systems Interconnection architecture, a set of semantic and syntactic rules that determine the behavior of entities in the same layer in performing communication functions. (T) (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. Synonymous with *line control discipline* and *line discipline*. See *bracket protocol* and *link protocol*.

**PSDN.** Packet switching data network.

**PSH.** Physical services header.

**PTF.** Program temporary fix.

**PU.** Physical unit.

**PU type.** (1) Deprecated term for *node type*. (2) The type of physical unit in a node.

**PU T2.1.** Deprecated term for *T2.1 node*.

**public network.** A network established and operated by a telecommunication Administration or by a Recognized Private Operating Agency (RPOA) for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. Contrast with *user-application network*.

**PUT.** Program update tape.

**PVC.** Permanent virtual circuit.

## Q

**QLLC.** Qualified logical link control.

**QOS.** Quality of service.

**qualified logical link control (QLLC).** An X.25 protocol that allows the transfer of data link control information between two adjoining SNA nodes that are connected through an X.25 packet-switching data



network. The QLLC provides the qualifier “Q” bit in X.25 data packets to identify packets that carry logical link protocol information.

**quality of service (QOS).** A set of communication characteristics required by an application. Each QOS defines a specific transmission priority, level of route reliability, and security level. Each QOS also defines whether the sessions are interactive. Contrast with *class of service (COS)*.

**queue.** (1) A list constructed and maintained so that the next data element to be retrieved is the one stored first. (T) (2) A line or list of items waiting to be processed; for example, work to be performed or messages to be displayed. (3) To arrange in or form a queue.

## R

**read-only memory (ROM).** Memory in which stored data cannot be modified by the user except under special conditions.

**receive not ready (RNR).** In communications, a data link command or response that indicates a temporary condition of being unable to accept incoming frames.

**receive not ready (RNR) packet.** See *RNR packet*.

**receive pacing.** In SNA, the pacing of message units that a component is receiving. Contrast with *send pacing*.

**RECFMS.** Record formatted maintenance statistics.

**Recognized Private Operating Agency (RPOA).** Any individual, company, or corporation, other than a government department or service, that operates a telecommunication service and is subject to the obligations undertaken in the Convention of the International Telecommunication Union and in the Regulations; for example, a communication common carrier.

**Recommendation X.21.** See *X.21*.

**Recommendation X.25.** See *X.25*.

**Recommendation X.28.** See *X.28*.

**Recommendation X.29.** See *X.29*.

**Recommendation X.3.** See *X.3*.

**recommended action.** Procedures suggested by the NetView program that can be used to determine the causes of network problems.

**record.** (1) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In

some programming languages, records are called structures. (l) (2) A set of data treated as a unit. (T) (3) A set of one or more related data items grouped for processing. (4) In VTAM, the unit of data transmission for record mode. A record represents whatever amount of data the transmitting node chooses to send.

**record formatted maintenance statistics (RECFMS).** A statistical record built by an SNA controller and usually solicited by the host.

**redirection.** The use of a logical device name that has been redirected so that data is obtained from a different source or is sent to a different destination.

**relation.** (1) In a relational database, a set of entity occurrences that have the same attributes. (T) (2) The comparison of two expressions to see if the value of one is equal to, less than, or greater than the value of the other. (3) In a relational database, a table that identifies entities and their attributes. Synonymous with *flat file*.

**release.** (1) A distribution of a new product or new function and APAR fixes for an existing product. Normally, programming support for the prior release is discontinued after some specified period of time following availability of a new release. The first version of a product is announced as Release 1, Modification Level 0. (2) In VTAM, to relinquish control of resources (communication controllers or physical units). See also *resource takeover*. Contrast with *acquire*.

**remote.** Pertaining to a system, program, or device that is accessed through a telecommunication line. Contrast with *local*. Synonym for *link-attached*.

**remove.** In the IBM Token-Ring Network, to take an attaching device off the ring.

**request unit (RU).** A message unit that contains control information, end-user data, or both.

**request/response unit (RU).** A generic term for a request unit or a response unit. See *request unit (RU)* and *response unit (RU)*.

**reset.** On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

**reset confirmation packet.** In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to inform the data circuit-terminating equipment (DCE) that a reset operation has been processed.

**reset request packet.** In X.25 communications, a packet transmitted by the data terminal equipment (DTE) to the data circuit-terminating equipment (DCE) to request that a virtual call or a permanent virtual circuit be reset. The reason for the request can also be specified in the packet.

## Glossary

**resource.** (1) Any facility of a computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In the NetView program, any hardware or software that provides function to the network.

**Resource Object Data Manager (RODM).** A component of the NetView program that operates as a cache manager and that supports automation applications. RODM provides an in-memory cache for maintaining real-time data in an address space that is accessible by multiple applications.

**resource resolution table (RRT).** In NPM, this table contains the names of network resources for which data is to be collected. The NPM RRT corresponds with an NCP and is built by NPMGEN from an NCP Stage I and an NCP RRT.

**resource takeover.** In VTAM, an action initiated by a network operator to transfer control of resources from one domain to another without breaking the connections or disrupting existing LU-LU sessions on the connection. See also *acquire* and *release*.

**resource types.** In the NetView program, a concept to describe the organization of panels. Resource types are defined as central processing unit, channel, control unit, and I/O device for one category; and communication controller, adapter, link, cluster controller, and terminal for another category. Resource types are combined with data types and display types to describe display organization. See also *data types* and *display types*.

**response.** In data communication, a reply represented in the control field of a response frame. It advises the primary or combined station of the action taken by the secondary or other combined station to one or more commands. See also *command*.

**response time.** (1) The elapsed time between the end of an inquiry or demand on a computer system and the beginning of the response; for example, the length of time between an indication of the end of an inquiry and the display of the first character of the response at a user terminal. (I) (A) (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

**response unit (RU).** A message unit that acknowledges a request unit. It may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to BIND SESSION). If negative, the response unit contains sense data defining the exception condition.

**reverse charging acceptance.** A facility that enables a data terminal equipment (DTE) to receive incoming packets that request reverse charging.

**REX.** Route extension.

**RNAA.** Request network address assignment.

**RNR.** Receive not ready.

**RNR packet.** A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate a temporary inability to accept additional packets for a virtual call or permanent virtual circuit.

**ROM.** Read-only memory. (A)

**route.** (1) An ordered sequence of nodes and transmission groups (TGs) that represent a path from an origin node to a destination node traversed by the traffic exchanged between them. (2) The path that network traffic uses to get from source to destination.

**route extension (REX).** In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *explicit route (ER)*, *path*, and *virtual route (VR)*.

**router.** (1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached. (4) Contrast with *bridge* and *gateway*.

**routing.** (1) The process of determining the path to be used for transmission of a message over a network. (T) (2) The assignment of the path by which a message is to reach its destination. (3) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

**RouTing update Protocol (RTP).** The Virtual NETworking System (VINES) protocol that maintains the routing database and allows the exchange of routing information between VINES nodes. See also *Internet Control Protocol (ICP)*.

**RPOA.** Recognized Private Operating Agency.

**RR.** Receive ready.

**RR packet.** A packet used by a data terminal equipment (DTE) or by a data circuit-terminating equipment (DCE) to indicate that it is ready to receive data packets within the window.

**RTP.** Routing update Protocol.

**RU.** Request/response unit.

**RU chain.** In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded. Each RU belongs to only one chain, which has a beginning and an end indicated by means of control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

## S

**SABM.** Set asynchronous balanced mode.

**scanner.** (1) A device that examines a spatial pattern one part after another, and generates analog or digital signals corresponding to the pattern. Scanners are often used in mark sensing, pattern recognition, or character recognition. (l) (A) (2) For the 3725 communication controller, a processor dedicated to controlling a small number of telecommunication lines. It provides the connection between the line interface coupler hardware and the central control unit.

**screen.** In Basic CUA architecture, the physical surface of a display device upon which information is shown to a user.

**SCS.** SNA character string.

**SDLC.** Synchronous Data Link Control.

**SDT.** Start data traffic.

**secondary logical unit (SLU).** In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU)*.

**secondary logical unit (SLU) key.** A key-encrypting key used to protect a session cryptography key during its transmission to the secondary half-session.

**segment.** (1) A section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a

combination of building cable and patch cables that are connected. (2) In Internet communications, the unit of transfer between TCP functions in different machines. Each segment contains control and data fields; the current byte-stream position and actual data bytes are identified along with a checksum to validate received data. (3) In IMS/VS, the unit of access to a database; for the database system, the smallest amount of data that can be transferred by one DL/I operation. For input terminal operations using the DC feature, a segment is defined by the particular terminal type and is obtained by the application program with one call. (4) Synonym for *BIU segment*. (5) See *link connection segment*.

**segmentation.** A process by which path control (PC) divides basic information units (BIUs) into smaller units, called BIU segments, to accommodate smaller buffer sizes in adjacent nodes. Both segmentation and segment assembly are optional PC features. The support for either or both is indicated in the BIND request and response.

**segmenting.** In OSI, a function performed by a layer to map one protocol data unit (PDU) from the layer it supports into multiple PDUs.

**select.** To explicitly identify one or more objects to which a subsequent choice will apply.

**send pacing.** In SNA, the pacing of message units that a component is sending. Contrast with *receive pacing*.

**sequence number.** (1) In communications, a number assigned to a particular frame or packet to control the transmission flow and receipt of data. (2) A numerical value assigned by VTAM to each message exchanged between two nodes. The value (one for messages sent from the application program to the logical unit and another for messages sent from the logical unit to the application program) increases by one for each successive message transmitted unless it is reset by the application program with a set and test sequence numbers (STSN) indicator.

**serial.** (1) Pertaining to a process in which all events occur one after the other; for example, serial transmission of the bits of a character according to V24 CCITT protocol. (T) (2) Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. (A) (3) Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. (A) (4) Contrast with *parallel*.

**Service Level Reporter (SLR).** A licensed program that generates management reports from data sets such as System Management Facility (SMF) files.

## Glossary

**service point (SP).** An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

**Service Point Application (SPA) Router.** In Communications Manager/2, a program that receives commands issued from the NetView program and sends the commands to an application program to be processed on an OS/2 workstation.

**session.** (1) In network architecture, for the purpose of data communication between functional units, all the activities which take place during the establishment, maintenance, and release of the connection. (T) (2) A logical connection between two network accessible units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) accompanying any transmissions exchanged during the session.

**session activation request.** In SNA, a request that activates a session between two network accessible units (NAUs) and specifies session parameters that control various protocols during session activity; for example, BIND and ACTPU. Contrast with *session deactivation request*.

**session control (SC).** In SNA, either of the following:

- One of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification.
- A request unit (RU) category used for requests and responses exchanged between the session control components of a session and for session activation and deactivation requests and responses.

**session deactivation request.** In SNA, a request that deactivates a session between two network accessible units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind*. Contrast with *session activation request*.

**session-level pacing.** A flow control technique that permits a receiving half-session or session connector to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *adaptive*

*session-level pacing, fixed session-level pacing, and virtual route (VR) pacing.*

**session parameters.** In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network accessible units. See also *logon mode*.

**session services (SS).** One of the types of network services in the control point (CP) and in the logical unit (LU). These services provide facilities for an LU or a network operator to request that a control point (an ENCP, NNCP, or SSCP) assist with initiating or terminating sessions between logical units. Assistance with session termination is needed only by SSCP-dependent LUs. See *configuration services, maintenance services, and management services*.

**set asynchronous balanced mode (SABM).** In communications, a data link control command used to establish a data link connection with the destination in asynchronous balanced mode.

**shared.** Pertaining to the availability of a resource for more than one use at the same time.

**shutdown.** The process of ending operation of a system or a subsystem, following a defined procedure.

**SLR.** Service Level Reporter.

**SLU.** Secondary logical unit.

**SMMF.** SSCP monitor mode function.

**SMP.** System Modification Program.

**SN.** Subarea node.

**SNA.** Systems Network Architecture.

**SNA character string (SCS).** In SNA, a character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

**SNA network.** The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network accessible units (NAUs), boundary function, gateway function, and intermediate session routing function components; and the transport network.

**SNA network interconnection (SNI).** The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

**SNAP.** Subnetwork Access Protocol.

**SNI.** SNA network interconnection.

**socket.** The abstraction provided by Berkeley Software Distribution (BSD) that serves as an endpoint for communication between processes or applications.

**softcopy.** (1) A nonpermanent copy of the contents of storage in the form of a display image. (T) (2) One or more files that can be electronically distributed, manipulated, and printed by a user. Contrast with *hardcopy*.

**solicited message.** A response from VTAM to a command entered by a program operator. Contrast with *unsolicited message*.

**SP.** Service point.

**span.** In the NetView program, a user-defined group of network resources within a single domain. Each major or minor node is defined as belonging to one or more spans. See also *span of control*.

**span of control.** The total network resources over which a particular network operator has control. All the network resources listed in spans associated through profile definition with a particular network operator are within that operator's span of control.

**SRT.** Symbol resolution table.

**SS.** (1) Start-stop. (2) Session services.

**SSCP.** System services control point.

**SSCP-dependent LU.** An LU that requires assistance from a system services control point (SSCP) in order to initiate an LU-LU session. It requires an SSCP-LU session.

**SSCP ID.** In SNA, a number that uniquely identifies a system services control point (SSCP). The SSCP ID is used in session activation requests sent to physical units (PUs) and other SSCPs.

**SSCP-LU session.** In SNA, a session between a system services control point (SSCP) and a logical unit (LU). The session enables the LU to request the SSCP to help initiate LU-LU sessions.

**SSCP monitor mode function (SMMF).** A function within NCP that keeps NCP resources active when an external SSCP has not established ownership of NCP.

**SSP.** System Support Programs.

**ST.** Session configuration screen abbreviation.

**stage.** A program that processes messages in a NetView pipeline. Stages send messages to each other serially.

**start-stop (SS) transmission.** (1) Asynchronous transmission such that each group of signals representing a character is preceded by a start signal and is followed by a stop signal. (T) (A) (2) Asynchronous transmission in which a group of bits is (a) preceded by a start bit that prepares the receiving mechanism for the reception and registration of a character, and (b) followed by at least one stop bit that enables the receiving mechanism to come to an idle condition pending reception of the next character.

**statement.** A language syntactic unit consisting of an operator, or other statement identifier, followed by one or more operands. See *definition statement*.

**static.** (1) In programming languages, pertaining to properties that can be established before execution of a program; for example, the length of a fixed length variable is static. (I) (2) Pertaining to an operation that occurs at a predetermined or fixed time. (3) Contrast with *dynamic*.

**station.** An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line.

**statistic.** In the NetView program, a resource-generated database record that contains recoverable error counts, traffic, and other significant data about a resource.

**status.** The condition or state of hardware or software, usually represented by a status code.

**status monitor.** A component of the NetView program that collects and summarizes information on the status of resources defined in a VTAM domain.

**stream.** (1) To send data from one device to another. (2) See *data stream*.

**subarea.** A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

**subarea address.** A value in the subarea field of the network address that identifies a particular subarea. See also *element address*.

**subarea link.** In SNA, a link that connects two subarea nodes. See also *channel link*.

## Glossary

**subarea network.** Interconnected subareas, their directly attached peripheral nodes, and the transmission groups that connect them.

**subarea node (SN).** A node that uses network addresses for routing and maintains routing tables that reflect the configuration of the network. Subarea nodes can provide gateway function to connect multiple subarea networks, intermediate routing function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes can be subarea nodes.

**subarea PU.** In SNA, a physical unit in a subarea node. Contrast with *peripheral PU*.

**Subnetwork Access Protocol (SNAP).** In LANs, a 5-byte protocol discriminator that identifies the non-IEEE standard protocol family to which a packet belongs. The SNAP value is used to differentiate between protocols that use \$AA as their service access point (SAP) value.

**suboperand.** One of multiple elements in a list comprising an operand. See also *definition statement*.

**subsystem.** A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system. (T)

**subvector.** A subcomponent of the NMVT major vector.

**supervisor call (SVC).** A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

**SVC.** (1) Supervisor call. (2) Switched virtual circuit.

**SWINN.** Switched intermediate network node.

**switched connection.** (1) A mode of operating a data link in which a circuit or channel is established to switching facilities as, for example, in a public switched network. (T) (2) A connection established by dialing. Contrast with *nonswitched connection*.

**switched intermediate network node (SWINN).** Deprecated term for *intermediate routing node*.

**switched line.** A telecommunication line in which the connection is established by dialing. Contrast with *nonswitched line*.

**switched major node.** In VTAM, a major node whose minor nodes are physical units and logical units attached by switched SDLC links.

**switched virtual circuit (SVC).** An X.25 circuit that is dynamically established when needed. The X.25 equiv-

alent of a switched line. Contrast with *permanent virtual circuit (PVC)*.

**synchronous.** (1) Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) (2) Occurring with a regular or predictable time relationship.

**Synchronous Data Link Control (SDLC).** (1) A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. (I) (2) Contrast with *binary synchronous communication (BSC)*.

**SYNTAX.** In the Simple Network Management Protocol (SNMP), a clause in the MIB module that defines the abstract data structure that corresponds to a managed object.

**system.** In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions. (I) (A)

**system configuration.** A process that specifies the devices and programs that form a particular data processing system.

**system definition.** The process, completed before a system is put into use, by which desired functions and operations of the system are selected from various available options. Synonymous with *system generation*.

**system generation.** Synonym for *system definition*.

**System Modification Program (SMP).** A program used to install software and software changes on MVS systems.

**system restart.** Synonym for *initial program load (IPL)*.

**system services control point (SSCP).** A component within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory services and other session services for end users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

**system services control point (SSCP) domain.** The system services control point, the physical units (PUs),

the logical units (LUs), the links, the link stations, and all the resources that the SSCP has the ability to control by means of activation and deactivation requests.

**system startup.** Synonym for *initial program load (IPL)*.

**System Support Programs (SSP).** An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

**Systems Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks. The layered structure of SNA allows the ultimate origins and destinations of information, that is, the end users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

## T

**TAB.** Terminal anchor block.

**table.** A repository for data that NETDA/2 uses to design a network. Each table contains information related to the network.

**takeover.** The process by which the failing active subsystem is released from its extended recovery facility (XRF) sessions with terminal users and replaced by an alternate subsystem. See *resource takeover*.

**task.** In a multiprogramming or multiprocessing environment, one or more sequences of instructions treated by a control program as an element of work to be accomplished by a computer. (I) (A)

**task panel.** Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See *help panel*.

**TCAM.** Telecommunications Access Method. Synonymous with *ACF/TCAM*.

**telecommunication line.** (1) The portion of a data circuit external to a data circuit-terminating equipment (DCE) that connects the DCE to a data-switching exchange (DSE), that connects a DCE to one or more other DCEs, or that connects a DSE to another DSE. (T) (2) Any physical medium, such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

**Telecommunications Access Method (TCAM).** An access method used to transfer data between main storage and remote or local terminals.

**teletypewriter exchange service (TWX).** Teletypewriter service in which suitably arranged teletypewriter stations are provided with lines to a central office for access to other such stations throughout the U.S. and Canada. Both baudot- and ASCII-coded machines are used. Business machines may also be used, with certain restrictions.

**terminal.** A device, usually equipped with a keyboard and a display device, that is capable of sending and receiving information.

**TG.** Transmission group.

**threshold.** (1) In the NetView program, a percentage value, set for a resource and compared to a calculated error-to-traffic ratio. (2) In NPM, high or low values supplied by the user to monitor data and statistics being collected. (3) In IBM bridge programs, a value set for the maximum number of frames that are not forwarded across a bridge due to errors, before a "threshold exceeded" occurrence is counted and indicated to network management programs. (4) An initial value from which a counter is decremented to 0, or a value to which a counter is incremented or decremented from an initial value.

**throughput class.** In packet switching, the speed at which data terminal equipment (DTE) packets travel through the packet switching network.

**Time Sharing Option (TSO).** An operating system option; for the System/370 system, the option provides interactive time sharing from remote terminals.

**time stamp.** (1) To apply the current system time. (2) The value on an object that is an indication of the system time at some critical point in the history of the object. (3) In query, the identification of the day and time when a query report was created that query automatically provides on each report.

**timeout.** (1) An event that occurs at the end of a predetermined period of time that began at the occurrence of another specified event. (I) (2) A time interval allotted for certain operations to occur; for example, response to polling or addressing before system operation is interrupted and must be restarted.

**TL.** Test loop.

**TLPB.** Transport-layer protocol boundary.

**TP.** (1) Transaction program. (2) Transmission priority.

**TRAC.** Trace record.

**trace.** (1) A record of the execution of a computer program. It exhibits the sequences in which the

## Glossary

instructions were executed. (A) (2) For data links, a record of the frames and bytes transmitted or received.

**transaction.** In IMS/VS, a specific set of input data that triggers execution of a specific process or job. A transaction is a message destined for an application program.

**transaction program (TP).** (1) A program that processes transactions in an SNA network. There are two kinds of transaction programs: application transaction programs and service transaction programs. See also *conversation*. (2) In VTAM, a program that performs services related to the processing of a transaction. One or more transaction programs may operate within a VTAM application program that is using the VTAM application program interface (API). In that situation, the transaction program would request services from the application program, using protocols defined by that application program. The application program, in turn, could request services from VTAM by issuing the APPCCMD macroinstruction.

**translation table.** A table used to replace one or more characters with alternative characters; for example, to translate characters representing a virtual address to those representing a real address, characters representing an event to those representing a procedure call, characters of a national character set to those of another national language, or characters representing a relocated address to those representing an absolute address.

**transmission group (TG).** (1) A connection between adjacent nodes that is identified by a transmission group number. See also *parallel transmission groups*. (2) In a subarea network, a single link or a group of links between adjacent nodes. When a transmission group consists of a group of links, the links are viewed as a single logical link, and the transmission group is called a *multilink transmission group (MLTG)*. A *mixed-media multilink transmission group (MMMLTG)* is one that contains links of different medium types (for example, token-ring, switched SDLC, nonswitched SDLC, and frame-relay links). (3) In an APPN network, a single link between adjacent nodes.

**transmission group (TG) profile.** In VTAM, a named set of characteristics (such as cost per byte, cost per unit of time, and capacity) that is used for APPN links.

**transmission group (TG) vector.** A representation of an endpoint TG in a T2.1 network, consisting of two control vectors: the TG Descriptor (X'46') control vector and the TG Characteristics (X'47') control vector.

**transmission header (TH).** Control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control

to route message units and to control their flow within the network. See also *path information unit*.

**transmission line.** Synonym for *telecommunication line*.

**transmission priority.** A rank assigned to a message unit that determines its precedence for being selected by the path control component in each node along a route for forwarding to the next node in the route.

**transmission services (TS) profile.** In SNA, a specification in a session activation request (and optionally, in the responses) of transmission control (TC) protocols (such as session-level pacing and the usage of session-level requests) to be supported by a particular session. Each defined transmission services profile is identified by a number.

**transmission subsystem component (TSC).** The component of VTAM that comprises the transmission control, path control, and data link control layers of SNA.

**transport-layer protocol boundary (TLPB).** In MPTN architecture, the protocol boundary that provides access in a protocol-independent fashion to multiple transport protocols.

**TSC.** (1) Telecommunications subsystem controller. (2) Transmission subsystem component.

**TSO.** Time Sharing Option.

**TWX.** Teletypewriter exchange service.

**type 2.1 node.** A node that can be an APPN network node, an APPN end node, or a LEN node. It can also attach as a peripheral node to a subarea boundary node in the same way as a type 2.0 node.

**T1.** In the United States, a 1.544-Mbps public access line. It is available in twenty-four 64-Kbps channels. The European version (E1) transmits 2.048 Mbps. The Japanese version (J1) transmits 1.544 Mbps.

## U

**UA.** Unnumbered acknowledgment.

**UNBIND.** In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

**unformatted.** In VTAM, pertaining to commands (such as LOGON or LOGOFF) entered by an end user and sent by a logical unit in character form. The character-coded command must be in the syntax defined in the user's unformatted system services definition table.



Synonymous with *character-coded*. Contrast with *field-formatted*.

**uninterpreted name.** In SNA, a character string that a system services control point (SSCP) can convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or Initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

**unsolicited message.** A message, from VTAM to a program operator, that is unrelated to any command entered by the program operator. Contrast with *solicited message*.

**UP.** Unnumbered poll.

**upstream.** In the direction of data flow from the end user to the host. Contrast with *downstream*.

**upstream device.** For the IBM 3710 Network Controller, a device located in a network such that the device is positioned between the IBM 3710 and a host. A communication controller upstream from the IBM 3710 is an example of an upstream device. Contrast with *downstream device*.

**user.** (1) A person who requires the services of a computing system. (2) Any person or any thing that may issue or receive commands and messages to or from the information processing system. (T) (3) Anyone who requires the services of a computing system.

**user-application network.** A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. (T) Contrast with *public network*.

**user exit.** (1) A point in an IBM-supplied program at which a user exit routine may be given control. (2) A programming service provided by an IBM software product that may be requested during the execution of an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

**user exit routine.** A user-written routine that receives control at predefined user exit points. User exit routines can be written in assembler or a high-level language.

**user table.** In TPNS, one or more text data entries contained in a table format, which may be referred to for logic testing and message generation.

## V

**V.24.** In data communications, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE).

**V.35.** In data communications, a specification of the CCITT that defines the list of definitions for interchange circuits between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) at various data rates.

**validation.** The checking of data for correctness or for compliance with applicable standards, rules, and conventions. (A)

**value.** (1) A specific occurrence of an attribute; for example, "blue" for the attribute "color." (T) (2) A quantity assigned to a constant, a variable, a parameter, or a symbol.

**vector.** The MAC frame information field.

**version.** A separately licensed program that usually has significant new code or new function.

**view.** In IBM network management products, a graphical representation of a network or a part of a network. A view consists of resource symbols and resource labels; it may also include a background picture or text that a network operator has placed within it.

**virtual circuit.** (1) In packet switching, the facilities provided by a network that give the appearance to the user of an actual connection. (T) See also *data circuit*. Contrast with *physical circuit*. (2) A logical connection established between two DTEs.

**virtual machine (VM).** In VM, a functional equivalent of a computing system. On the 370 Feature of VM, a virtual machine operates in System/370 mode. On the ESA Feature of VM, a virtual machine operates in System/370, 370-XA, ESA/370, or ESA/390 mode. Each virtual machine is controlled by an operating system. VM controls the concurrent execution of multiple virtual machines on an actual processor complex.

**Virtual Machine/Enterprise Systems Architecture (VM/ESA).** An IBM licensed program that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a real machine.

**Virtual Machine/Extended Architecture (VM/XA).** An operating system that facilitates conversion to MVS/XA by allowing several operating systems (a production system and one or more test systems) to run simultaneously on a single 370-XA processor. The VM/XA Migration Aid has three components: the control

## Glossary

program (CP), the conversational monitor system (CMS), and the dump viewing facility.

**Virtual Machine/System Product (VM/SP).** An IBM licensed program that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a real machine.

**virtual route (VR).** In SNA, either a) a logical connection between two subarea nodes that is physically realized as a particular explicit route or b) a logical connection that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual route pacing, and provides data integrity through sequence numbering of path information units (PIUs). Contrast with *explicit route (ER)*. See also *path* and *route extension (REX)*.

**virtual route (VR) pacing.** In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *session-level pacing*.

**Virtual Storage Extended (VSE).** An IBM licensed program whose full name is the Virtual Storage Extended/Advanced Function. It is a software operating system controlling the execution of programs.

**Virtual Telecommunications Access Method (VTAM).** An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

**VM.** Virtual machine.

**VM/ESA.** Virtual Machine/Enterprise Systems Architecture.

**VM/SP.** Virtual Machine/System Product.

**VM/XA.** Virtual Machine/Extended Architecture.

**VM/370 control program (CP).** The component of VM/370 that manages the resources of a single computer with the result that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of an IBM System/370 computing system.

**VP.** Variable pageable.

**VR.** Virtual route.

**VSE.** Virtual Storage Extended. Synonymous with *VSE/Advanced Functions*.

**VSE/Advanced Functions.** The basic operating system support needed for a VSE-controlled installation. Synonym for *VSE*.

**VTAM.** (1) Virtual Telecommunications Access Method. (2) Synonymous with *ACF/VTAM*.

**VTAM definition.** The process of defining the user application network to VTAM and modifying IBM-defined characteristics to suit the needs of the user.

**VTAM definition library.** The operating system files or data sets that contain the definition statements and start options filed during VTAM definition.

**VTAM operator command.** A command used to monitor or control a VTAM domain. See also *definition statement*.

## W

**wideband.** Synonym for *broadband*. (T)

**window.** (1) A portion of a display surface in which display images pertaining to a particular application can be presented. Different applications can be displayed simultaneously in different windows. (A) (2) An area with visible boundaries that presents a view of an object or with which a user conducts a dialog with a computer system. (3) In data communication, the number of data packets a data terminal equipment (DTE) or data circuit-terminating equipment (DCE) can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing, or flow control, of packets. (4) See *pacing window*.

**window size.** The specified number of frames of information that can be sent before receiving an acknowledgment response.

## X

**X-on.** Transmitter on.

**X.21.** An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for a general-purpose interface between data terminal equipment and data circuit-terminating equipment for synchronous operations on a public data network.

**X.25.** An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. See also *packet switching*.

**X.25 interface.** An interface consisting of a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE) in communication over a link using the procedures described in the CCITT Recommendation X.25.

**X.25 NCP Packet Switching Interface (NPSI).** An IBM licensed program that allows SNA users to communicate over packet switching data networks that have interfaces complying with CCITT Recommendation X.25. It allows SNA programs to communicate with SNA or non-SNA equipment over such networks.

**X.25\_1984.** CCITT Recommendation X.25 (Malaga-Torremolinos, 1984).

**X.28.** An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for the DTE/DCE interface for a start-stop mode data terminal equipment (DTE) accessing the packet assembly/disassembly (PAD) facility in a public data network situated in the same country.

**X.29.** An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for procedures for the exchange of control information and user data between a packet assembly/disassembly (PAD) facility and a packet mode data terminal equipment (DTE) or another PAD facility.

**X.3.** An International Telegraph and Telephone Consultative Committee (CCITT) recommendation for packet assembly/disassembly (PAD) in a public data network.

**X.32.** An International Telegraph and Telephone Consultative Committee (CCITT) recommendation that defines the interface between data terminal equipment (DTE) and packet-switching networks through a public switched network, such as a public telephone network.

**XA.** Extended architecture.

**XI.** X.25 SNA Interconnection.

**XID.** Exchange identification.



---

## Bibliography

---

### X.25 NCP Packet Switching Interface Publications

The following paragraphs briefly describe the library of books associated with X.25 NCP Packet Switching Interface Version 3.

*X.25 NPSI Version 3 General Information* (GC30-3469)

This book introduces managers, system designers, programmers, and other data processing personnel to the basic concepts of packet-switching, X.25, and IBM's X.25 NCP Packet Switching Interface licensed program.

*X.25 NPSI Version 3 Release 9 Data Areas* (LY30-5627)

This book is written for system programmers to assist them in troubleshooting and diagnosing problems with the X.25 NCP Packet Switching Interface. It provides control block information for NPSI.

*NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning* (LY30-5610)

This book is written for system programmers to assist them in troubleshooting and diagnosing problems with the X.25 NCP Packet Switching Interface. It helps programmers to diagnose problems, resolve common errors, and describe problems to and interface with the IBM Support Center.

*X.25 NPSI Version 3 Host Programming* (SC30-3502)

This book is written for application and system programmers to assist them in writing application programs that use the X.25 NCP Packet Switching Interface. Application programmers should have some knowledge of DATE and GATE functions and the operating systems that support them. System programmers should be knowledgeable in SNA architecture.

*NCP Version 7 and X.25 NPSI Version 3 Planning and Installation* (SC30-3470)

This book provides system programmers and analysts with the information required to plan and implement the installation of NPSI. The topics include hardware and software requirements, preinstallation system performance considerations, instructions for defining and generating NPSI, and installation examples.

---

### NCP, SSP, and EP Library

The following publications apply to the NCP, SSP, and EP library.

*NCP and EP Reference* (LY43-0029)

*NCP and EP Reference Summary and Data Areas* (LY43-0030)

*NCP and SSP Customization Guide* (LY43-0031)

*NCP and SSP Customization Reference* (LY43-0032)

*NCP, SSP, and EP Diagnosis Guide* (LY43-0033)

*NCP, SSP, and EP Generation and Loading Guide* (SC31-6221)

*NCP, SSP, and EP Messages and Codes* (SC31-6222)

*NCP, SSP, and EP Resource Definition Guide* (SC31-6223)

*NCP, SSP, and EP Resource Definition Reference* (SC31-6224)

*NCP, SSP, and EP Trace Analysis Handbook* (LY43-0037)

*NCP V7R8.1 Migration Guide* (SC30-4024)

*NCP V7R8.1, SSP V4R8.1, and EP R14 Library Directory* (SC30-4025)

*NTuneNCP Feature Reference* (LY43-0039)

*NTuneMON User's Guide* (SC31-6266)

---

### Other Networking System Product Libraries

The following sections summarize the libraries for cross-product information, SNA Services (VTAM), NetView, and NPM.

### Networking Library

The following books provide cross-product information for NCP, NetView, and VTAM. For detailed information about these products, refer to the library for each.

*ACF/NCP, ACF/SSP, EP, NPSI, and NTuneMON Softcopy Collection Kit* (LK2T-0414, CD-ROM)

*IBM Online Libraries: Softcopy Collection Kit User's Guide* (GC28-1700)

*Planning for Integrated Networks* (SC31-8062)

*Planning for NetView, NCP, and VTAM* (SC31-8063)

### SNA Services Library (formerly VTAM Library)

The following list shows the books in the SNA Services library.

*OS/390 SecureWay Communications Server: SNA Planning and Migration Guide* (SC31-8622)

*OS/390 SecureWay Communications Server: SNA Network Implementation* (SC31-8563)

*OS/390 SecureWay Communications Server: SNA Resource Definition Reference* (SC31-8565)

*OS/390 SecureWay Communications Server: SNA Operation* (SC31-8567)

*OS/390 SecureWay Communications Server: Quick Reference* (SX75-0121)

*OS/390 SecureWay Communications Server: SNA Diagnosis Vol 1* (LY43-0079)

*OS/390 SecureWay Communications Server: SNA Diagnosis Vol 2* (LY43-0080)

*OS/390 SecureWay Communications Server: SNA Messages* (SC31-8569)

*OS/390 SecureWay Communications Server: IP and SNA Codes* (SC31-8571)

*OS/390 SecureWay Communications Server: Data Areas Vol 1* (LY43-0111)

*OS/390 SecureWay Communications Server: Data Areas Vol 2* (LY43-0112)

*OS/390 SecureWay Communications Server: SNA Customization* (LY43-0110)

*OS/390 SecureWay Communications Server: Anynet Sockets over SNA* (SC31-8577)

### NetView Library

The following list shows the books in the NetView library.

*TME 10 NetView for OS/390 NGMF User's Guide* (SC31-8234)

*TME 10 NetView for OS/390 User's Guide* (SC31-8241)

*TME 10 NetView for OS/390 Administration and Security Reference* (SC31-8222)

*TME 10 NetView for OS/390 Application Programmer's Guide* (SC31-8223)

*TME 10 NetView for OS/390 Automation Guide* (SC31-8225)

*TME 10 NetView for OS/390 Bridge Implementation* (SC31-8238)

*TME 10 NetView for OS/390 Command Reference* (SC31-8227)

*TME 10 NetView for OS/390 Customization Guide* (SC31-8228)

*TME 10 NetView for OS/390 Customization: Using Assembler* (SC31-8229)

*TME 10 NetView for OS/390 Customization: Using PL/I and C* (SC31-8230)

*TME 10 NetView for OS/390 Customization: Using REXX and the NetView Command List Language* (SC31-8231)

*TME 10 NetView for OS/390 Installation and Administration Guide* (SC31-8236)

*TME 10 NetView for OS/390 Messages* (SC31-8237)

*TME 10 NetView for OS/390 Diagnosis Guide* (LY43-0108)

*TME 10 NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide* (SC31-8233)

*TME 10 NetView for OS/390 Tuning Guide* (SC31-8240)

*TME 10 NetView for OS/390 NGMF User's Guide*  
(SC31-8234)

## NPM Library

The following list shows the books in the NPM library.

*NPM Concepts and Planning* (GH19-6961-03)

*NPM Installation and Customization* (SH19-6964-03)

*NPM Desk/2 User's Guide* (SH19-6963-02)

*NPM User's Guide* (SH19-6962-03)

*NPM Messages and Codes* (SH19-6966-03)

*NPM Graphic Subsystem* (SH19-6967-00)

*NPM Reference* (SH19-6965-03)

*NPM Diagnosis* (LY19-6381-03)

---

## Related Publications

The following publications, though not directly related to this book, may be helpful in understanding your network.

## SNA Publications

The following publications contain information on SNA:

*Systems Network Architecture Formats* (GA27-3136)

*Systems Network Architecture: Management Services Reference* (SC30-3346)

*Systems Network Architecture Technical Overview*  
(GC30-3073)

*X.25 DTE/DCE and DTE/DTE Interface Architecture Reference* (SC30-3409)

*X.25 1980 Interface for Attaching IBM SNA Nodes to Packet-Switched Data Networks: General Information Manual* (GA27-3345)

*X.25 1984/1988 DTE/DCE and DTE/DTE Interface General Information Manual* (GA27-3761)

## 372x Publications

The following list shows selected publications for the IBM 372x.

*3720/3721 Communication Controllers Configuration Guide* (GA33-0063)

*3720/3721 Communication Controllers Introduction*  
(GA33-0060)

*3725 Communication Controllers Principles of Operation*  
(GA33-0013)

## 3745 Publications

The following list shows the publications for the IBM 3745.

*IBM 3745 Basic Operations Guide for MOSS Operations* (SA33-0098)

*IBM 3745 Communication Controller Introduction*  
(GA33-0092 for 3745 Models 210, 310, 410 and 610)

*IBM 3745 Communication Controller Introduction*  
(GA33-0138 for 3745 Models 130, 150, and 170)

*IBM 3745 Communication Controller (all models): Principles of Operation* (SA33-0102)

*IBM 3745 Models 21A through 61A Migration and Planning Guide* (GA33-0183)

## CCITT Recommendations

The following book provides information about CCITT recommendations X.1–X.32.

*Blue Book, Volume VIII—Fascicle VIII.2, Data Communication Networks: Services and Facilities, Interfaces, Recommendations X.1–X.32.*

## GOSIP

The following book provides information about the United States Government Open System Interconnection Profile (GOSIP).

*Federal Information Processing Standard (FIPS) 146-2*

## Bibliography

### ISO Publications

The following books provide information about ISO standards 7776 and 8208.

*ISO 7776 Information Processing Systems—Data Communication—High-Level Data Link Control Procedures—Description of the X.25 LAPB-Compatible DTE Data Link Procedures*

*ISO 8208 Information Processing Systems—Data Communication—X.25 Packet Level Protocol for Data Terminal Equipment.*



# Index

## Numerics

- 256 Kbps Support 3-8
- 3174 3-26
- 3710 6-9
- 3720
  - CF3720 configurator 3-55
  - compatibility 3-18
- 3745
  - CF3745 configurator 3-55
  - compatibility 3-18
- 3746 Model 900
  - ADDRESS keyword considerations 5-50
  - overview 3-19
  - support of X.21 leased connections 3-23
  - support of X.21 switched connections 3-24
  - support of X.25 connections 3-18
  - support of X.25 ODLC 1-2
- 4361 4-8
- 7820 3-21
- 9370 4-7

## A

- access method
  - described 3-45
  - dial-out 7-7
  - hardware and software compatibility 3-18
  - in relation to connectivity status 7-5
  - in relation to fast connect 3-32
  - simulation of call-out 4-31
- ACCOUNT keyword 5-51
- activation
  - calls that use SVCSC 7-4
  - DATE 7-9
  - GATE 7-7
  - PVC 7-2, 7-6
  - SVC 7-3, 7-6
- ACTIVTO keyword 4-7, 5-51
- ACTPU command 7-2, 7-9
- ADDIFG keyword on X25.MCH statement 5-51
- ADDR keyword 6-6
- address field 3-4
- ADDRESS keyword 5-49
- ADJLINK keyword
  - X25.CPL statement 5-15
  - X25.FCG statement 5-24
  - X25.LINE statement 5-33
  - X25.LU statement 5-43
  - X25.MCH statement 5-51
  - X25.PU statement 5-92
  - X25.VC statement 5-100

- ADJNODE keyword
  - X25.CPL statement 5-16
  - X25.FCG statement 5-24
  - X25.LINE statement 5-33
  - X25.LU statement 5-43
  - X25.MCH statement 5-52
  - X25.PU statement 5-92
  - X25.VC statement 5-101
- AEFBYTE keyword 5-52
- AEFCTCP keyword 5-53
- AGGPRI keyword
  - X25.CPL statement 5-16
  - X25.FCG statement 5-25
  - X25.LINE statement 5-33
  - X25.LU statement 5-43
  - X25.MCH statement 5-53
  - X25.PU statement 5-92
  - X25.VC statement 5-101
- ALTCTCP keyword 5-15
- ANS keyword
  - X25.CPL statement 5-17
  - X25.FCG statement 5-25
  - X25.MCH statement 5-53
  - X25.VC statement 5-102
- application attachment
  - non-SNA resources 7-7
  - SNA resources 7-5
- application level 3-47
- applications, invoking by X.25 virtual circuit
  - number 4-9
- ASCII 3-46, 5-77

## B

- BAK prefix 5-6
- BAL error messages E-1
- BAL prefix 5-6
- billing information reports 4-34
- binary synchronous communication (BSC) 2-7, 2-10
- BIND parameter 3-47, 7-5
- BNN 3-27, 5-35
  - See also* peripheral node (BNN)
- bracket protocol 3-46
- BRKCON keyword 4-7, 5-54
- buffer pools 6-19
- buffer size 3-6, 3-45
- BUILD definition statement
  - keywords
    - NPA 5-8
    - SESSACC 5-8
    - X25.1MWINDOW 5-9
    - X25.BYTTHROD 5-8
    - X25.IDNUMH 5-8

## Index

### BUILD definition statement (*continued*)

#### keywords (*continued*)

X25.MAXPIU 5-8  
X25.MCHCNT 5-9  
X25.MMODULO 5-9  
X25.OUTIDNUM 5-9  
X25.PADINDX 5-9  
X25.PREFIX 5-2, 5-9  
X25.SEGTHROD 5-10  
X25.SNAP 5-10  
X25.USGTIER 5-10

overview 5-8

table of keywords C-2

## C

CALL ACCEPTED command 4-15, A-9

Call Accepted packet

description 3-41

dial-in for non-SNA resources 7-6

selection of packet and window sizes 4-15

sent by DATE CTCP 4-25

SVC setup 7-10

call collision

avoiding 7-8

under SVC 4-6

Call Connected packet

facilities checking 3-10, A-3

used for communication between DTEs A-6

CALL keyword

for definition of restricted virtual circuit A-7

for generation of DATE support 3-39, 4-30

for PATH statement definition 6-15

on X25.LINE statement 5-33

on X25.VC statement 5-102

to perform call-in 4-32

to perform call-out 4-32

with fast connect 3-36

Call Redirection facility 4-17

CALL REQUEST command

LLC specification for call-in 3-41

LLC specification for call-out 3-41

Call Request packet

call-ins 4-24

call-outs 4-26

PCNE-to-PCNE considerations 4-41

call user data byte 0 (CUD0) 3-40

call-in

keywords for call-in definitions 4-8

LLC-type selection 3-40

on a virtual circuit 4-32

PATH statement definition for SVCSC 6-16

simulated call-in for GATE 7-7

virtual circuit connection parameters for 4-15

call-inout 6-17

call-out

LLC type selection 3-41

PATH statement definition for SVCSC 6-15

simulated by call-in for DATE 4-31

simulated by call-in for GATE 4-32, 7-7

SVC selection for 4-26

with GATE and DATE 4-16

called address extension facility (AEF) A-5

called address field 4-24

calling DTE address verification 4-17

Casual Connection

CCXDELAY keyword 5-54

description 4-14

migration considerations 3-51

cause and diagnostic table 3-10

CAUSE keyword

effect on cause and diagnostic codes 3-10

X25.MCH statement 5-54

X25.NET statement 5-81

CCITT

facilities supported by NPSI A-10

Recommendation X.25 3-4, 3-5

CCITT 84 diagnostic codes 3-10

CCITT keyword 5-80, 5-81

CCXDELAY 5-54

CD-ROM documentation xiv

central control unit (CCU) utilization

calculating 3-55

effect of isolated pacing response on 3-45, 6-19

effect of packet size on 3-5

effect of packet-level window size on 3-6

Character String mode 3-45

characteristics, physical and electrical 3-4

characters (special) 6-9

CICS

adjustments 3-47

resource naming conventions 5-2

Clear Confirmation packet 7-10

Clear Indication packet A-6

Clear Request packet A-6

sent because of format error 3-11

sent for DATE deactivation 7-10

with fast select option A-6

closed user group (CUG)

consideration when using SVCSC 4-17

description 3-6, A-5

CLRSESSQ keyword

X25.CPL statement 5-17

X25.FCG statement 5-25

X25.MCH statement 5-54

X25.VC statement 5-102

CLSDST command 7-4, 7-10

CMTKYWRD keyword

X25.CPL statement 5-17

X25.FCG statement 5-25

X25.LINE statement 5-34

- CMTKYWRD keyword (*continued*)
    - X25.LU statement 5-43
    - X25.MCH statement 5-54
    - X25.PU statement 5-92
    - X25.VC statement 5-102
  - coding examples (NPSI in NCP)
    - generation input and JCL B-3
    - LKEDT step B-12
    - main switched major node B-22
    - NCP input by NDF B-13
    - password protection and LU 6.2 B-2, B-30
    - SMN for SNA type 2.1 and SVCSC B-27
  - COMMITO keyword 5-34, 5-103
  - communication and transmission control program
    - See CTCP
  - communication controller
    - physical circuits for 4-2
    - planning for the configurator 3-55
    - port address 4-2
    - supported by NPSI 3-18
  - configurators 3-55
  - CONNECT keyword 5-54
  - connectibility status
    - non-SNA PVC 7-6
    - non-SNA SVC 7-7
    - SNA PVC 7-3
    - SNA SVC 7-5
  - connections
    - non-SNA 3-29
    - SNA 3-26, 7-4
    - without a PSDN 3-20
  - CONTACT command 7-6
  - continuous service 3-44
  - control blocks
    - generated under MVS 2-5
    - generated under VM 2-10
    - generated under VSE 2-12
    - generation macros 5-6
    - PU name relationship for NCP and VTAM 4-6
  - control packet 3-4
  - correspondence table
    - for LLC type selection 3-41, 4-24
    - used for call-in 4-32
  - CPHINDX keyword 5-82
  - CPNAME
    - call-in definition 4-9
    - keyword on PU statement 6-7, 6-14
    - SVC definition 4-6, 4-14
  - CTCP
    - communication with DATE
      - definitions 4-30
      - described 3-38, 7-9
      - for call-out 4-26
    - communication with GATE
      - definitions 4-31
      - described 3-31, 7-7
  - CTCP (*continued*)
    - control of reverse charging acceptance A-8
    - control of virtual circuits 4-23
    - facilities support A-10
    - fast connect selection of 4-33
    - multiple interfacing 3-35
    - positional parameters 4-33
    - selection using Address Extension Facility 4-29
    - selection using CUD0 4-24, 4-28
    - using subaddressing for selection 4-25
  - CTCP keyword 3-35, 5-55
  - CTCPNO keyword
    - X25.CPL statement 5-15
    - X25.FCG statement 4-34, 5-23
    - X25.LINE statement 5-34
    - X25.VC statement 5-103
  - CUD 4-16, 4-41
  - CUD0 keyword
    - CONNECT keyword value 5-55
    - used to determine LLC type 4-28, 4-31
    - X25.MCH statement 5-56
  - Customer Information Control System
    - See CICS
- ## D
- D-bit 3-47, 5-57
  - data circuit-terminating equipment
    - See DCE
  - data packets 3-4
  - data set ready (DSR) 7-2, 7-9
  - Data Stream Compatibility (DSC) mode 3-45
  - data terminal equipment
    - See DTE
  - data terminal ready (DTR) 7-2, 7-9
  - DBIT keyword 5-57
  - DCE 3-4, 3-20
  - DCI keyword 5-82, E-32
  - DCINx keyword 5-95
  - DCOTx keyword 5-96
  - deactivation
    - calls that use SVCSC 7-5
    - DATE 7-10
    - PVC 7-3, 7-6
    - SVC 7-4
  - Dedicated access to X.25 transport extension (DATE)
    - considerations 3-10, 7-9
    - CTCP definitions 4-30
    - described 3-38, 4-30
    - facilities support A-10
    - fast select A-6
    - LLC type selection 4-25
  - definite response 3-47
  - DEFPRI keyword
    - X25.LINE statement 5-34
    - X25.MCH statement 5-58

## Index

- DEFPRI keyword (*continued*)
    - X25.VC statement 5-103
  - delayed logical link activation 7-11
  - destination node 3-20
  - diagnostic codes 3-10, 3-20
  - dial out by PSDN E-22, E-33
  - dial-in connection 7-6
  - dial-out connection 3-29, 7-6
  - DIALNO keyword
    - for activating calls that use SVCSC 7-4
    - for call-out specification
      - on SVCSC statement 6-15
      - on X25.OUFT statement 4-16
      - on X25.VCCPT statement 4-16
    - for LLC type selection 3-41
    - for LLCI on X25.MCH statement 5-63
    - L values 4-27
    - on PATH definition statement 6-8
    - SVC selection for call-out 4-26
  - DIRECT keyword 5-58
  - DISCNT keyword 6-6
  - DLCADDR keyword
    - coding 6-11
    - L values 4-27
    - PATH definition statement for outgoing calls 6-8
    - using with DIALNO 6-12
  - DM keyword 5-81
  - domain, multiple 4-41
  - DSABLTO keyword 5-58
  - DSTNODE keyword 4-6, 5-35
  - DTE
    - closed user group A-5
    - link level 3-4
    - logical channel 4-2
    - network address 6-8
    - types 4-23
    - use of logical channel identifier 4-3
  - DTETODTE keyword 5-46, 5-58
  - dynamic NETID 4-18
  - dynamic reconfiguration 5-43, 5-93
- ## E
- EBCDIC 3-46, 5-77
  - enable presentation (ENP) characters 4-40
  - ENABLTO keyword 5-59
  - exception response 3-47
- ## F
- facility checking
    - nonstandard facilities 3-11
    - rejection of facilities 3-11
    - X25.NET statement
      - NSTFDAC 3-11
      - RFAC 3-11
  - facility codes 3-11, A-1
  - fast connect
    - configuration 4-36
    - considerations 7-8
    - described 3-32, 4-33
    - FCG keywords 4-34
    - MCH keywords 4-33
    - with two CTCPs 3-35
  - fast select 3-11, A-6
  - fast transaction processing interface
    - See FTPI
  - FASTRUN keyword 2-5, 2-9
  - FID3 transmission header 7-5
  - flow control parameter
    - negotiation 4-15, A-2
    - values A-9
  - frame 3-4
  - frame check sequence 3-4
  - frame reject (FRMR) 3-9
  - FRMLGTH keyword 5-50
  - FTPI
    - considerations for non-SNA resources 7-8
    - defined 3-36
    - keyword on X25.MCH statement 4-39, 5-59
  - FTPITH keyword on X25.MCH statement 5-60
  - FTPITHTO keyword on X25.MCH statement 5-60
- ## G
- GATE keyword
    - described 3-31
    - to generate DATE support 3-39
    - to generate fast connect support 3-35
    - X25.MCH statement 5-60
  - GATEINFO keyword on X25.MCH statment 5-60
  - GENEND definition statement 5-12
  - general access to X.25 transport extension (GATE)
    - considerations 3-10, 7-7
    - described 3-31, A-10
    - function 4-31, A-6
  - generation process of NPSI
    - input including JCL B-2
    - ISO impacts 3-9
    - resource names 5-2
    - under MVS 2-4—2-7
    - under VM 2-8—2-10
    - under VSE 2-10—2-12
  - GROUP statements 5-3, 6-16
  - GRPNM keyword 6-15
- ## H
- hardware supported 3-18
  - HEXNAME keyword
    - X25.FCG statement 4-34, 5-25
    - X25.VC statement 5-103

- high-priority class of traffic 3-7, A-6
  - host program considerations 3-46
  - host resident programs 3-18
  - HPRQLIM keyword 5-92, 5-103
  - HPTSS processing 3-19
  - HPTSS support 5-50
  - hung resource 7-3, 7-10
  - hunt group 4-17
- I**
- IBM00 table 3-10
  - IBM80 table 3-10
  - ID keyword 5-35, 5-104
  - IDBLK keyword 4-9, 4-13
  - IDBLK, relationship with LCGN 4-9
  - IDBLKC keyword
    - examples of use 4-13
    - migration considerations 3-51
    - not used with IDBLKLCG 5-61
    - X25.MCH statement 4-14, 5-60
  - IDBLKG keyword
    - examples of use 4-13
    - migration considerations 3-51
    - not used with IDBLKLCG 5-61
    - X25.MCH statement 4-14, 5-60, 5-61
  - IDBLKLCG keyword 4-9, 5-61
  - IDBLKP keyword
    - examples of use 4-13
    - migration considerations 3-51
    - not used with IDBLKLCG 5-61
    - X25.MCH statement 4-14
  - identification number 4-9, 4-41
  - IDNUM keyword
    - call-in definitions 4-8
    - call-out definitions 4-13
    - described 4-9, 6-15
    - for SVCSC 4-17, 6-14
    - PCNE considerations 4-41
    - to locate PU definition in an SMN 6-14
  - IDNUMH keyword 4-9
  - IDNUMT keyword 5-35, 5-104
  - IMS 3-47, 5-2
  - INACTO keyword 5-82
  - Incoming Call packet
    - conditions for acceptance A-9
    - for fast select A-3
    - for flow control parameter negotiation A-8
    - for reverse charging A-6
    - NPSI facility checking 3-7, 3-10
    - PCNE-to-PCNE considerations 4-41
    - specification of values 5-63
  - INDEX keyword
    - X25.OUFT statement 5-87
    - X25.PAD statement 5-89
    - X25.VCCPT statement 5-111
  - inhibit presentation (INP) characters 4-40
  - INSLOW keyword 5-112, 6-19
  - installation
    - NPSI under MVS 2-4
    - NPSI under VM 2-8, 2-11
  - integrated PAD
    - See PAD (packet assembler/disassembler), integrated
  - International Standards Organization (ISO) compliance
    - modifications at link level
      - V3 3-9
      - V3R4 3-14
      - V3R5 3-16
    - modifications at packet level
      - V3 3-9
      - V3R4 3-15
      - V3R5 3-16
    - modifications at system generation 3-9, 3-14
  - Interrupt confirmation 3-46
  - Interrupt packet 7-10
  - Interrupt request 3-46
  - INTFAC keyword 5-61
  - INTWLINK keyword 5-61
  - IPL 3-28, 6-4
  - IPL keyword 5-35
  - ISDN 3-21, B-117
  - ISTATUS keyword
    - LU definition statement 6-4
    - X25.CPL statement 5-17
    - X25.FCG statement 5-26
    - X25.LINE statement 5-35
    - X25.MCH statement 5-62
    - X25.VC statement 5-104
  - ITRACE keyword 5-62
- L**
- LAP 5-50, 5-62
  - LAPB
    - frame window size used by 5-9
    - process initiation 7-2, 7-10
  - LCGDEF keyword 4-3, 5-50
  - LCGN keyword 4-4, 5-30
  - LCN keyword
    - X25.LINE statement 4-4, 5-35
    - X25.VC statement 4-4, 5-99
  - LCN0 keyword 5-62
  - legal notices ix
  - levels of X.25
    - link level
      - standard protocols 3-4
      - window size 3-6
    - packet level
      - described 3-4
      - window size 3-6
    - physical level 3-4

## Index

- libraries
    - MVS 2-4
    - VM 2-8
    - VSE 2-11, 2-12
    - VTAMLST source 4-6
  - line speed 3-8, A-8
  - link access procedure
    - See LAP
  - link access procedure balanced
    - See LAPB
  - link-edit step 2-4, 2-8
  - linkage editor
    - input under MVS 2-5
    - input under VM 2-9
    - input under VSE 2-12
    - overview 2-2
  - LKEDT step B-2
  - LLC
    - extended (one-for-one correspondence) 4-28
    - selection
      - defined 3-40, 4-23
      - PVC 3-40, 4-24
      - SVC 3-40, 4-24
      - user-defined table example 4-30
      - using CUD0 3-40, 4-24
    - types 3-26
    - using DATE 3-38, 3-41, 4-25
    - using GATE 3-31, 4-31
    - using subaddressing 4-24
  - LLC keyword 5-36, 5-104
  - LLC0 keyword 5-63
  - LLC2 keyword 5-63
  - LLC3 keyword 5-63
  - LLC4 keyword 5-63
  - LLC5 keyword 5-63
  - LLC1 keyword 5-63
  - LLCLIST keyword
    - for GATE support 4-33
    - for PAD support 4-40
    - for SVCSC 4-8
    - X25.MCH statement 5-62
  - LLERP keyword 5-93, 5-105
  - load modules
    - entry point prefix 5-6
    - MVS 2-5
    - VM 2-9
    - VSE 2-12
  - LOCADDR keyword 5-43
  - log mode entry table 4-34
  - LOGAPPL keyword
    - for fast connect option 3-33, 4-34
    - SVC call-out 4-27
    - X25.MCH statement 5-63
  - logical channel
    - described 3-5, 4-2
    - figure 4-2
  - logical channel (*continued*)
    - identifier 3-5, 4-2
    - number 4-4, A-7
  - LOSTERM 7-10
  - LSPRI keyword 5-64
  - LU (logical unit) 3-31
  - LU definition statement 6-4, 6-14
  - LU simulator 3-29, 7-6
  - LU type 6.2 6-14, B-1, B-2
  - LUDR keyword 5-43
  - LUDRPOOL definition statement 6-2
  - LUNAME keyword
    - for fast connect option 4-33
    - X25.CPL statement 5-17
    - X25.MCH statement 5-64
- ## M
- MACB keyword 5-64
  - macros shipped with NPSI D-1
  - MAXDATA keyword
    - for PIU segmentation 3-45
    - on LU definition statement 6-4
    - on X25.FCG statement 5-26
    - on X25.PU statement 5-93
    - on X25.VC statement 5-105
  - MAXOUT keyword 5-93, 5-105
  - MAXPIU keyword 5-8
  - MAXPKTL keyword 5-111, A-7
  - MAXTIME keyword 5-17
  - MBITCHN keyword 3-42, 5-64
  - MBNOFULL keyword 5-64
  - MCH lockups 7-4
  - MCHCNT keyword 5-9
  - migration
    - V1 to V3 releases 3-49
    - V3R2 to V3R3 3-51
    - V3R3 to V3R4 3-51
    - V3R4 to V3R5 3-52
    - V3R5 to V3R6 3-53
    - V3R6 to V3R7 3-53
    - V3R7 to V3R8 3-53
    - V3R8 to V3R9 3-54
  - MINDATA keyword 5-17
  - MMODULO keyword 5-9, 5-65
  - MODE keyword 5-36
  - modem eliminator 3-20
  - MODETAB keyword
    - for fast connect option 4-34
    - X25.CPL statement 5-18
    - X25.MCH statement 5-65
  - modulo 128 5-50, A-5
  - modulo 8 5-50, A-5
  - MONLINK keyword 5-36
  - more data bit (M-bit) A-6

multichannel link (MCH) 3-31, 4-32, 5-75, B-2  
 MWINDOW keyword 5-9, 5-50

## N

naming conventions 5-2, 5-6  
 NCP buffer size (BFRS) 3-45  
 NCP customization facility 2-4, 2-8  
 NCP definition statements  
   LUDRPOOL 6-2  
   OPTIONS 6-2  
 NCP Packet Switching Interface (NPSI) 2-2  
 NCP X.25 ODLC  
   See X.25 ODLC (Outboard Data Link Control)  
 NCP/EP Definition Facility (NDF) 2-2, 2-4, 2-8  
 NCPGRP keyword  
   for call-out 6-16  
   on X25.CPL statement 5-18  
   on X25.FCG statement 5-26  
   on X25.LINE statement 5-37  
   on X25.MCH statement 5-65  
   on X25.VC statement 5-105  
 NDF standard attachment facility 2-7, 2-10  
 NDRETRY keyword 5-65  
 NETID keyword 6-15  
   SVCS considerations 4-17, 6-15  
   X25.MCH statement 5-65  
 NETTYPE keyword 5-82  
 NetView performance monitor  
   See NPM  
 NetView program 4-6  
 Network Control Program (NCP)  
   generation with NPSI  
     migration 3-49  
     under MVS 2-2  
     under VM 2-8  
     under VSE 2-10  
   statement changes 6-2  
   statements created by NPSI 6-3, B-2  
 Network Interface Adapter (NIA) 3-26  
 network planning  
   host application program considerations  
     CICS, IMS, TSO adjustments 3-46  
     non-SNA communication 3-46  
     transparent PAD implementation 3-46  
   network resource recovery considerations  
     short hold mode function 3-26  
     switched session continuation 3-48  
   SNA considerations 3-45  
 Network Terminal Option (NTO) 2-5  
 NEWDEFN keyword 6-2  
 NGFINC keyword  
   X25.CPL statement 5-18  
   X25.FCG statement 5-26  
   X25.LINE statement 5-37  
   X25.LU statement 5-43

NGFINC keyword (*continued*)  
   X25.MCH statement 5-66  
   X25.PU statement 5-93  
   X25.VC statement 5-106  
 NGFOPT keyword  
   X25.CPL statement 5-18  
   X25.FCG statement 5-27  
   X25.LINE statement 5-37  
   X25.LU statement 5-43  
   X25.MCH statement 5-66  
   X25.PU statement 5-93  
   X25.VC statement 5-106  
 NGFTEXT keyword  
   X25.CPL statement 5-19  
   X25.FCG statement 5-27  
   X25.LINE statement 5-37  
   X25.LU statement 5-43  
   X25.MCH statement 5-67  
   X25.PU statement 5-93  
   X25.VC statement 5-107  
 non-SNA connections 3-29  
 non-SNA resources  
   PVCs 7-6  
   SVCs 7-6  
 nonstandard facility processing 3-11  
 NOTOPCS keyword 5-37  
 NPA keyword 5-8  
 NPACOLL keyword 3-49, 5-67  
 NPADTEAD keyword 5-68  
 NPM 3-48  
   accounting 3-48, 4-19  
   performance 3-49, 4-20  
 NPM publications X-45  
 NPPVCN keyword 5-68  
 NPRETRY keyword 5-68  
 NRZI mode 5-45  
 NSTDFAC keyword 3-11, 5-83  
 NUMILU keyword 6-2  
 NUMTYP1 keyword 6-2  
 NUMTYP2 keyword 6-2

## O

ODLC keyword 5-47, 5-68  
 one-way logical channel A-7  
 Open Systems Interconnection (OSI) 3-32  
 operating system 3-18  
 OPTFACL keyword 3-7, 5-87, A-7  
 optional user facilities  
   supported by NPSI A-9  
   table of A-10  
 OPTIONS definition statement  
   coding for NPSI 6-2  
   used for generation input verification  
     for MVS 2-5  
     for VM 2-9  
     for VSE 2-11

## Index

OSI priority 3-42  
OSITYPE keyword  
  for PVC 4-24  
  X25.LINE statement 5-38  
  X25.VC statement 5-107  
OUFINDX keyword 5-38, 5-108  
OUHINDX keyword 5-83  
OWNER keyword 5-68

## P

  pacing  
    parameters 3-45  
    virtual route 6-19  
  PACING keyword 3-47  
  packet  
    described 3-4  
    header 3-5, 4-2  
    length 3-5  
    routing 3-5, 4-2  
    size  
      defined to PSDN 3-5  
      defining A-2  
      maximum 4-15  
      negotiation facility field 3-11, 3-13  
    window size A-2  
      defining 3-6, 4-15  
      link level 3-6  
      negotiation facility field  
      packet level 3-6  
  packet switched data network (PSDN)  
    cost determination 3-7  
    defining facilities A-1  
    described 3-4  
    planning subscription 3-5, 4-30  
    virtual circuit connection  
      logical channels 4-2  
      PVC 3-5  
      SVC 3-5  
    without connection to 3-20  
  PAD (packet assembler/disassembler)  
    coding PAD keyword  
    integrated  
      described 3-8  
      implementation 4-40  
      with DATE 4-30  
    SDLC 4-40  
    service 3-8  
    support 3-37, 4-40, 6-14  
    transparent  
      described 3-8  
      implementation 3-46, 4-40  
      with DATE 3-39  
      with USSTAB and GATE  
  PAD keyword  
    for LLC type 5 3-37  
  PAD keyword (*continued*)  
    X25.MCH statement 4-40, 5-69  
  PADBRKCD keyword 5-69  
  PADCHECK keyword 5-69  
  PADINDX keyword 5-69  
  PADPARM keyword 5-69, 5-89  
  password protection 4-40, 5-70, B-1  
  PATH definition statement  
    correlation of 6-17  
    overview 4-14, 4-41  
    SVCSC considerations 4-17, 6-15, 7-4  
  path information unit (PIU) 3-5, 3-38, 7-5  
  PCNE-to-PCNE communication 4-41  
  peak traffic periods 3-34  
  peripheral node (BNN) 3-26  
  permanent virtual circuit (PVC)  
    described 3-5  
    in virtual circuit definition 4-5  
    LLC type selection 3-40  
    non-SNA resources 7-6  
    selection 4-24  
    set up 7-10  
    SNA resources 7-2  
    specifications 4-31  
  physical circuit 4-2, 4-30, 4-32  
  Physical Services Header (PSH) 3-26, 7-2  
  PHYSRSC keyword 5-70  
  piggybacking  
    link level 3-44, 3-55  
    packet level 3-44, 3-55  
    packet window size effect on 3-6  
  PIU segmentation 3-31, 3-45  
  PKTMODL keyword 5-70  
  planning for NPSI 3-45  
  PLPIGGYB keyword 5-70  
  preface xi  
  PREFIX keyword 5-9  
  presentation level 3-47  
  PRFLINE keyword  
    X25.FCG statement 4-34, 5-27  
    X25.VC statement 5-108  
  PRFLU keyword  
    X25.FCG statement 4-34, 5-28  
    X25.VC statement 5-108  
  PRFPU keyword  
    X25.FCG statement 4-34, 5-28  
    X25.VC statement 5-108  
  PSDN E-15, E-22  
  PU definition statement  
    *See also* X25.PU statement  
    in VTAM 4-14, 6-7  
    keywords  
      BRKCON 4-7, 5-51  
      IDBLK 4-13, 6-7  
      IDNUM 4-13, 4-17, 6-7  
      MAXDATA 6-7  
      MAXOUT 6-7



- PU definition statement (*continued*)
    - keywords (*continued*)
      - NETID 4-17
      - PUTYPE 4-8, 4-13, 5-65, 6-7
      - SUBAREA 4-17
    - overview 6-3, 6-6
    - specifications for SVCSC 4-8, 4-17
  - PU type 2 SNA host 3-26
  - publications
    - NPM X-45
    - related X-45
    - SecureWay X-44
    - SNA X-44
    - VTAM X-44
  - PUDR keyword 5-93
  - PUNAME keyword
    - PU statement 6-3
    - X25.CPL statement 5-19
    - X25.MCH statement 5-70
  - PUTYPE keyword
    - call-in definitions 4-8
    - call-out definitions 4-13
    - PU statement 4-8
    - SNA type 2.1 nodes
    - use of NETID keyword with 5-65
    - X25.PU statement 5-93
  - PWPROT keyword 5-70
- Q**
- Q-bit 3-26, 3-46
  - QOS
    - priority definition 3-43, 4-36
    - range definition 3-43
    - support for 3-43
  - QTY keyword 4-34, 5-23
  - qualified data bit
    - See Q-bit
  - qualified data packet 7-10
  - Qualified Logical Link Control (QLLC) 3-26, 3-27, 7-2
  - quality of service facility
    - See QOS
- R**
- R20 keyword 3-12, 5-85
  - R22 keyword 3-12, 5-85
  - R23 keyword 3-12, 5-86
  - receive ready (RR) transmission 3-44
  - Recommendation X.21
    - See X.21 Recommendation
  - Recommendation X.21bis
    - See X.21bis Recommendation
  - Recommendation X.25
    - See X.25 Recommendation
  - Recommendation X.28
    - See X.28 Recommendation
  - Recommendation X.29
    - See X.29 Recommendation
  - Recommendation X.3
    - See X.3 Recommendation
  - recovery 5-38
  - relationship between LCGN and IDBLOCK 4-9
  - relay program 3-31
  - REQUEST CONTACT command
  - request/response unit
    - See RU
  - reset confirmation 3-46
  - Reset packet
    - considerations 3-14
    - retries 3-12
  - RESETINO keyword 5-83
  - RESETPVC keyword
    - on X25.MCH statement 5-70
    - on X25.NET statement 5-84
  - RESETUNB keyword
    - on X25.LINE statement 5-38
    - on X25.MCH statement 5-71
    - on X25.NET statement 5-84
    - on X25.VC statement 5-108
  - resource assignment process 7-11
  - resource control
    - non-SNA 7-5
    - SNA 7-2
  - Resource Resolution Table 2-6, 2-10
  - response time 3-8
  - Restart packet 3-12
  - retry counts 3-12
  - RETVCCCT keyword 5-38, 5-108
  - RETVCTO keyword 5-38, 5-108
  - reverse charging 3-7, A-7
  - RFAC
    - closed user group 3-6
    - facility checking 3-11
    - for closed user group A-6
    - for fast select A-6
    - for throughput class negotiation A-8
    - keyword 3-11
    - keyword on X25.NET statement 5-85
  - RNRPKT keyword 5-71
  - RNRTIMER keyword
    - on X25.MCH statement 5-71
    - on X25.NET statement 5-85
  - ROLE keyword 5-71
  - RPADPARM keyword 5-71
  - RU 3-31
  - RU Chaining 3-42
  - RUSIZE keyword 3-47

## Index

### S

- SABM (set synchronous balanced mode) 3-9, 3-15
- SDLC
  - LLC type 2 and 3 3-26, 7-2
  - PU statement 6-3
- SDRTCNT keyword 5-72
- SDRTIME keyword 5-72
- SecureWay publications X-44
- SEND command 3-47
- SESSACC keyword 5-8
- session continuation 3-48, 6-18, 7-7
- set asynchronous balanced mode
  - See SABM (set synchronous balanced mode)
- SHM keyword 5-72
- short hold mode (SHM)
  - described 4-6, 7-4
  - SHM keyword 4-7
  - SHM parameter 4-6, 4-17, 6-15, 7-4
  - SHMTIM keyword 6-15
  - with call-out using SVCSC 6-15
- SHUTD keyword 5-72
- SIMLOGON statement 7-3
- SLODOWN keyword 5-112, 6-19
- slowdown 3-55, 4-15
- slowdown parameters 4-16
- SNA (systems network architecture)
  - resources 2-2
- SNA network interconnect (SNI) 2-7, 2-10, 5-65
- SNA publications X-44
- SNA to non-SNA connections 3-26
- SNA to SNA connections 3-26
- SNA type 2.1
  - CPNAME 6-7
  - for LUDRPOOL definition statement 6-2
  - LLC type 3 connections 3-26
  - node 6-14
  - node definition 4-14
  - switched major node B-2, B-27
- SNAP trace facility 5-10
- SNI (SNA network interconnect) 2-7, 2-10, 5-65
- softcopy documentation xiv
- software release compatibility 3-18
- SPAN keyword
  - X25.CPL statement 5-19
  - X25.FCG statement 5-28
  - X25.LINE statement 5-39
  - X25.MCH statement 5-72
  - X25.VC statement 5-109
- SPEED keyword
  - piggybacking 3-44, 3-55
  - X25.MCH statement 5-72
- SPNQLLC keyword 5-74
- SRT thresholds 6-4
- start data traffic (SDT) 5-93, 6-7
- start-stop definitions 2-7, 2-10
- STATION keyword 5-74
- STATOPT keyword
  - X25.CPL statement 5-19, 5-28
  - X25.FCG statement 5-28
  - X25.LINE statement 5-39
  - X25.LU statement 5-43
  - X25.MCH statement 5-74
  - X25.PU statement 5-93
  - X25.VC statement 5-109
- storage requirements 3-18
- SUBADDR keyword 5-63, 5-75
- subaddressing field of Incoming Call packets 4-24, 4-31, 5-63
- subaddressing for CTCP 4-25
- subaddressing for LLC type 4-24, 4-32, 4-33
- SUBAREA keyword 4-17
- subarea node
  - DSTNODE keyword 5-35
  - LLC type 3 connection to 3-26, 3-28
- SUBD keyword 5-55, 5-75
- SUFFIX keyword
  - keyword 5-5, 5-25, 5-29
  - X25.FCG statement 4-34, 5-29
  - X25.VC statement 5-109
- SVCINN keyword 5-75
- switched major node (SMN)
  - description 4-6, 4-8, 7-3, B-2
  - PU statement 7-4
- switched virtual circuit subarea communication (SVCSC)
  - calling DTE address 4-17
  - closed user group (CUG) 4-17
  - initial activation 3-28
  - PATH statement definition
    - call in 6-16
    - call inout 6-17, 7-4
    - call out 4-17, 6-15, 7-4
    - correlation of 6-17
- PU statement keywords
  - IDNUM 4-17, 6-14, 6-15
  - NETID 4-17, 6-15
  - PUTYPE 6-14, 6-15
  - SUBAREA 4-17, 6-14
  - TGN 6-14
- support
  - activating calls 7-4
  - deactivating calls 7-5
- X25.MCH statement keywords
  - ACTIVTO 4-7
  - SDRTCNT 4-6, 5-72
  - SDRTIME 4-6, 5-72
  - SHM 4-6, 5-72
  - SVCINN 4-6, 5-75
- switched virtual circuits
  - described 7-3
  - establishment 4-6

- switched virtual circuits (*continued*)
  - LLC type selection 3-40
  - selection 4-24
  - setup 7-10
  - specifications 4-31
  - supported on PSDNs 3-5
- SWMMTID keyword 5-39
- SWMTT definition statement 5-13
- synchronization 3-46, 7-6
- system initialization 3-33
- system services control point (SSCP)
  - inactivity timer mechanism 4-7
  - recovery consideration 3-48
  - session continuation 6-18, 7-7

## T

- T1TIMER keyword 3-44, 5-76
- T2 timer 3-44
- T20 keyword 3-12, 5-86
- T21 keyword 3-12, 5-86
- T22 keyword 3-12, 5-86
- T23 keyword 3-12, 5-86
- T25 keyword 5-86
- T26 keyword 5-86
- T3 keyword 5-76
- T4 keyword
  - usage 3-9, 3-15
  - X25.MCH statement 5-76
- table assembly
  - cause and diagnostic 3-10
  - of NCP generation 2-8
  - virtual circuit parameters 4-15
- TADDR keyword 5-39
- TAXUNIT keyword 4-34, 5-76
- TDTIMER keyword 5-76
- Telecommunication Subsystem Controller (TSC) 4-7
- Terminal Adapter (TA), IBM 7820 3-21
- TGN keyword 5-93, 6-14
- throughput class negotiation A-8
- timer values 3-12
- TPTIMER keyword 5-76
- trademarks x
- TRAN keyword 5-77
- TRANSFR keyword 5-39, 5-77
- TSO 3-46
- TYPE keyword
  - with fast connect option 5-33
  - with SVCSC 5-99
  - X25.LINE statement 5-33
  - X25.VC statement 3-36, 5-99

## U

- unnumbered acknowledgment (UA) 3-9

- upstream address (USA) 6-9
- USER keyword 5-95
- USERGEN keyword 6-2
- USRFIL2 keyword 5-88
- USRFILE keyword 5-87
- USSTAB keyword 6-14

## V

- V net,DIAL command
  - after inactivity timeout 4-7, 5-51
  - with SHM 7-4
- V net,HANGUP command 7-5
- validity check
  - for generation under MVS 2-5
  - for generation under VM 2-9
  - of received packets 3-12
- VC to MCH mapping 4-18
- VCCINDX keyword
  - X25.LINE statement 4-15, 5-39
  - X25.MCH statement 5-77
  - X25.VC statement 5-100
- VCID keyword 5-78
- VERID keyword 6-16
- VERIFY keyword 4-17, 6-16
- videotex terminal 3-34
- virtual call 3-4, A-8
- virtual circuit
  - See also* LLC
  - described
    - PVC 4-5, 7-2
    - SVC 4-6, 7-3
  - selection 3-40
  - supported by NPSI
    - with CTCP A-9
    - without CTCP A-9
  - supported in PSDN 3-5
  - using DATE
    - activation 7-9
    - description 3-38, 4-30
    - network requirements 3-38
  - using fast connect 3-32, 7-8
  - using GATE
    - description 3-32
    - network requirements 4-31
    - processing considerations 7-7
- virtual circuit types
  - See also* LLC
  - LLC type 0 3-30, 4-25, 4-27, 4-28
  - LLC type 2 3-26
  - LLC type 3 3-27
  - LLC type 4 3-31, 4-31
  - LLC type 5 3-37
- virtual route pacing window size (VRPWS) 6-19
- VMODTAB keyword 4-34, 5-78

## Index

VPACING parameter 3-55, 6-4  
VRPWS keyword 6-19  
VTAM coding considerations  
  for SVCs 4-6  
  PATH statement 4-17, 6-7  
VTAM publications X-44  
VTAM statements  
  created by NDF 2-4  
  LLC type selection 3-41  
  PATH statement 4-26  
  PU statement for SVCSC 6-14  
VTAM User-Exit routine 4-17  
VTAM, definition of term xii  
VTAMLST source library 4-6  
VTMKYWRD keyword  
  X25.CPL statement 5-20  
  X25.FCG statement 5-29  
  X25.LINE statement 5-40  
  X25.LU statement 5-43  
  X25.MCH statement 5-78  
  X25.PU statement 5-93  
  X25.VC statement 5-110  
VWINDOW keyword 5-112

## W

WAIT option 3-47  
window  
  link level  
    default size 3-6  
    described 3-6  
  packet level  
    default size 3-6  
    described 3-6

## X

X.21  
  connections 3-19  
  enhanced support 4-22, 5-79  
  switched connections 3-24, 5-79, 7-8  
X.21 Recommendation 3-4  
X.21bis Recommendation 3-4  
X.25 ISDN interworking B-117  
X.25 NCP Packet Switching Interface  
  described 2-2, 3-3  
  generation  
    input including JCL B-2, B-3  
    input to VTAM B-2  
    ISO impacts 3-9  
    under MVS 2-4  
    under VM 2-5, 2-8  
    under VSE 2-10  
  installation  
    under MVS 2-4  
    under VM 2-8  
    under VSE 2-11

X.25 NCP Packet Switching Interface (*continued*)  
  installation examples B-3  
  migration  
    V1 to V3 releases 3-49  
    V3R2 to V3R3 3-51  
    V3R3 to V3R4 3-51  
    V3R4 to V3R5 3-52  
    V3R5 to V3R6 3-53  
    V3R6 to V3R7 3-53  
    V3R7 to V3R8 3-53  
  planning  
    considerations 3-18, 3-45  
    overview 3-3  
    support of CCITT facilities A-10  
X.25 ODLC (Outboard Data Link Control)  
  description 1-2, 2-2  
  generation  
    sample statements B-125  
    under MVS 2-4—2-8  
    under VM 2-8—2-10  
    under VSE 2-10—2-12  
  installation 1-3  
  keywords  
    CCITT 5-80, 5-81  
    DTETODTE 5-46, 5-58  
    ODLC 5-47, 5-68  
  migration considerations 3-49—3-54  
  planning for network 3-45—3-49  
  relationship with NPSI 2-2  
  sample generation statements B-125  
X.25 Recommendation 3-4, 3-5  
X.28 Recommendation 3-8  
X.29 Recommendation 3-8  
X.3 Recommendation 3-8  
X21INACT keyword 5-78  
X21INTWK keyword 5-79  
X21RTYCT keyword 5-79  
X21RTYTO keyword 5-79  
X21SW keyword 5-79  
X21SW keyword on X25.MCH statement 5-78  
X25.BYTTHROD keyword 5-8  
X25.CPL statement  
  keywords  
    ADJLINK 5-15  
    ADJNODE 5-16  
    AGGPRI 5-16  
    ALTCTCP 5-15  
    ANS 5-17  
    CLRSESSQ 5-17  
    CMTKYWRD 5-17  
    CTCPNO 5-15  
    ISTATUS 5-17  
    LUNAME 5-17  
    MAXTIME 5-17  
    MINDATA 5-17  
    MODETAB 5-18  
    NCPGRP 5-18

X25.CPL statement (*continued*)keywords (*continued*)

NGFINC 5-18  
 NGFOPT 5-18  
 NGFTXT 5-19  
 PUNAME 5-19  
 SPAN 5-19  
 STATOPT 5-19, 5-28  
 VTMKYWRD 5-20

table of keywords C-3

## X25.END statement 5-21, C-4

## X25.FCG statement

## keywords

ADJLINK 5-24  
 ADJNODE 5-24  
 AGGPRI 5-25  
 ANS 5-25  
 CLRSESSQ 5-25  
 CMTKYWRD 5-25  
 CTCENO 5-23  
 HEXNAME 5-25  
 ISTATUS 5-26  
 MAXDATA 5-26  
 NCPGRP 5-26, 6-15  
 NGFINC 5-26  
 NGFOPT 5-27  
 NGFTXT 5-27  
 PRFLINE 4-34, 5-27  
 PRFLU 4-34, 5-28  
 PRFPU 4-34, 5-28  
 QTY 5-23  
 SPAN 5-28  
 STATOPT 5-28  
 SUFFIX 4-34, 5-29  
 VTMKYWRD 5-29

overview 4-34, 5-22

resource naming conventions 5-4

table of keywords C-4

## X25.IDNUMH keyword 5-8

## X25.LCG statement

## keywords

LCGN 4-3, 5-30  
 table of C-5

overview 5-30

## X25.LINE statement

## keywords

ADJLINK 5-33  
 ADJNODE 5-33  
 AGGPRI 5-33  
 CALL 4-30, 5-33  
 CMTKYWRD 5-34  
 COMMITO 5-34  
 CTCENO 5-34  
 DEFPRI 5-34  
 DSTNODE 5-35  
 ID 5-35  
 IDNUMT 5-35

X25.LINE statement (*continued*)keywords (*continued*)

IPL 5-35  
 ISTATUS 5-35  
 LCN 4-4, 5-35  
 LLC 5-36  
 MODE 5-36  
 MONLINK 5-36  
 NCPGRP 5-37  
 NGFINC 5-37  
 NGFOPT 5-37  
 NGFTEXT 5-37  
 NOTOPCS 5-37  
 OSITYPE 4-24, 5-38  
 OUFINDX 5-38  
 RESETUNB 5-38  
 RETVCCT 5-38  
 RETVCTO 5-38  
 SPAN 5-39  
 STATOPT 5-39  
 SWMMTID 5-39  
 TADDR 5-39  
 TRANSFR 5-39  
 TYPE 5-33  
 VCCINDX 4-15, 5-39  
 VTMKYWRD 5-40

overview 5-31

table of keywords C-6

## X25.LU statement

## keywords

ADJLINK 5-43  
 ADJNODE 5-43  
 AGGPRI 5-43  
 CMTKYWRD 5-43  
 LOCADDR 5-43  
 LUDR 5-43  
 NGFINC 5-43  
 NGFOPT 5-43  
 NGFTXT 5-43  
 STATOPT 5-43  
 VTMKYWRD 5-43

overview 5-41

table of keywords C-7, C-8

## X25.MAXPIU keyword 5-8

## X25.MCH statement

## keywords

ACCOUNT 5-51  
 ACTIVTO 5-51  
 ADDIFG 5-51  
 ADDRESS 5-49  
 ADJLINK 5-51  
 ADJNODE 5-52  
 AEFBYTE 5-52  
 AEFCTCP 5-53  
 AGGPRI 5-53  
 ANS 5-53  
 BRKCON 5-54

## Index

### X25.MCH statement (*continued*)

#### keywords (*continued*)

CAUSE 5-54  
CCXDELAY 5-54  
CLRSESSQ 5-54  
CMTKYWRD 5-54  
CONNECT 4-33, 5-54  
CTCP 4-28, 5-55  
CUD0 4-28, 5-55, 5-56  
DBIT 5-57  
DEFPRI 5-58  
DIRECT 5-58  
DSABLTO 5-58  
DTETODTE 5-46, 5-58  
ENABLTO 5-59  
FRMLGTH 5-50  
FTPPI 5-59  
FTPITH 5-60  
FTPITHTO 5-60  
GATE 5-60  
GATEINFO 5-60  
IDBLKC 5-60  
IDBLKG 5-60  
IDBLKLCG 5-61  
IDBLKP 5-61  
INTFAC 5-61  
INTWLINK 5-61  
ISTATUS 5-62  
ITRACE 5-62  
LCGDEF 4-3, 5-50  
LCN0 5-62  
LLC0 5-63  
LLC2 5-63  
LLC3 5-63  
LLC4 5-63  
LLC5 5-63  
LLCI 5-63  
LLCLIST 4-33, 5-62  
LOGAPPL 5-63  
LSPRI 5-64  
LUNAME 5-64  
MACB 5-64  
MBITCHN 5-64  
MBSNOFULL 5-64  
MMODULO 5-9, 5-65  
MODETAB 5-65  
MWINDOW 5-50  
NCPGRP 5-65  
NDRETRY 5-65  
NETID 5-65  
NGFINC 5-66  
NGFOPT 5-66  
NGFTEXT 5-67  
NPACOLL 3-49, 5-67  
NPADTEAD 5-68  
NPPVCN 5-68  
NPRETRY 5-68

### X25.MCH statement (*continued*)

#### keywords (*continued*)

ODLC 5-47, 5-68  
OWNER 5-68  
PAD 4-40, 5-69  
PADBRKCD 5-69  
PADCHECK 5-69  
PADINDX 5-69  
PHYSRSC 5-70  
PKTMODL 5-70  
PLPIGGYB 5-70  
PUNAME 5-70, 6-3  
PWPROT 5-70  
RESETPVC 5-70  
RESETUNB 5-71  
RNRPKT 5-48, 5-71  
RNRTIMER 5-71  
ROLE 5-71  
RPADPARM 5-71  
SDRTCNT 4-6, 5-72  
SDRTIME 4-6, 5-72  
SHM 5-72  
SHUTD 5-72  
SPAN 5-72  
SPEED 3-55, 5-72  
SPNQLLC 5-74  
STATION 5-74  
STATOPT 5-74  
SUBADDR 5-75  
SUBD 5-55, 5-75  
SVCINN 4-6, 5-75  
T1TIMER 3-55, 5-76  
T3 5-76  
T4 5-76  
TAXUNIT 5-76  
TDTIMER 5-76  
TPTIMER 5-76  
TRAN 5-77  
TRANSFR 5-77  
VCCINDX 5-77  
VCID 5-78  
VMODTAB 5-78  
VTMKYWRD 5-78  
X21INACT 5-78  
X21INTWK 5-79  
X21RTYCT 5-79  
X21RTYTO 5-79  
X21SW 5-79  
XMONLNK 5-78  
overview 5-45  
table of keywords C-8  
X25.MCHCNT keyword 5-9  
X25.MMODULO keyword 5-9  
X25.MWINDOW keyword 5-9  
X25.NET statement  
CCITT 5-80, 5-81

X25.NET statement (*continued*)

## keywords

CAUSE 3-10, 5-81  
 CPHINDX 5-82  
 DCI 5-82  
 DM 5-81  
 INACTO 5-82  
 NETTYPE 5-82  
 NSTDFAC 3-11, 5-83  
 OUHINDX 5-83  
 R20 3-12, 5-85  
 R22 3-12, 5-85  
 R23 3-12, 5-86  
 RESETINO 5-83  
 RESETPVC 5-84  
 RESETUNB 5-84  
 RFAC 3-11, 5-85  
 RNRTIMER 5-85  
 T20 3-12, 5-86  
 T21 3-12, 5-86  
 T22 3-12, 5-86  
 T23 3-12, 5-86  
 T25 5-86  
 T26 5-86

overview 5-80

table of keywords C-11

## X25.OUFT statement

## keywords

INDEX 5-87  
 OPTFACL 5-87  
 USRFIL2 5-88  
 USRFILD 5-87

overview 4-16, 5-87

table of keywords C-12

## X25.OUTIDNUM keyword 5-9

## X25.PAD statement

## keywords

INDEX 5-89  
 PADPARAM 5-89

overview 5-89

table of keywords C-12

## X25.PADINDX keyword 5-9

## X25.PREFIX keyword 5-9

## X25.PU statement

## keywords

ADJLINK 5-92  
 ADJNODE 5-92  
 AGGPRI 5-92  
 CMTKYWRD 5-92  
 HPRQLIM 5-92  
 LLERP 5-93  
 MAXDATA 5-93  
 MAXOUT 5-93  
 NGFINC 5-93  
 NGFOPT 5-93  
 NGFTXT 5-93  
 PUDR 5-93

X25.PU statement (*continued*)keywords (*continued*)

PUTYPE 5-93  
 STATOPT 5-93  
 TGN 5-93  
 VTMKYWRD 5-93  
 XID 5-94

overview 5-90

table of keywords C-12

## X25.SEGTHROD keyword 5-10

## X25.SNAP keyword 5-10

## X25.TRAN statement

## keywords

DCINx 5-95  
 DCOTx 5-96  
 USER 5-95

overview 5-95

table of keywords C-15

## X25.USGTIER keyword 5-10

## X25.VC statement

## keywords

ADJLINK 5-100  
 ADJNODE 5-101  
 AGGPRI 5-101  
 ANS 5-102  
 CALL 4-30, 5-102  
 CLRSESSQ 5-102  
 CMTKYWRD 5-102  
 COMMITO 5-103  
 CTCPCNO 5-103  
 DEFPRI 5-103  
 HEXNAME 5-103  
 HPRQLIM 5-103  
 ID 5-104  
 IDNUMT 5-104  
 ISTATUS 5-104  
 LCN 4-4, 5-99  
 LLC 5-104  
 LLERP 5-105  
 MAXDATA 5-105, 6-4  
 MAXOUT 5-105  
 NCPGRP 5-105  
 NGFINC 5-106  
 NGFOPT 5-106  
 NGFTEXT 5-107  
 OSITYPE 4-24, 5-107  
 OUFINDX 5-108  
 PRFLINE 5-108  
 PRFLU 5-108  
 PRFPU 5-108  
 RESETUNB 5-108  
 RETVCCT 5-108  
 RETVCTO 5-108  
 SPAN 5-109  
 STATOPT 5-109  
 SUFFIX 5-109  
 TYPE 5-99

## Index

X25.VC statement (*continued*)  
  keywords (*continued*)  
    VCCINDX 4-15, 5-100  
    VTMKYWRD 5-110  
  overview 5-97  
  table of keywords C-16  
X25.VCCPT statement  
  keywords  
    INDEX 5-111  
    INSLOW 5-112  
    MAXPKTL 5-111  
    VWINDOW 5-112  
  overview 4-15, 5-111  
  table of keywords C-18  
XID 4-6, 4-9, 6-9  
XID keyword 4-14, 5-94  
XMONLNK keyword 5-78



---

# Tell Us What You Think!

**Network Control Program  
X.25 Planning and Installation  
NCP Version 7 and X.25 NPSI Version 3  
Publication No. SC30-3470-13**

We hope you find this publication useful, readable, and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form. If you are in the USA, you can mail this form postage free or fax it to us at 1-800-253-3520. Elsewhere, your local IBM branch office or representative will forward your comments or you may mail them directly to us.

|                                                                          |                          |                          |
|--------------------------------------------------------------------------|--------------------------|--------------------------|
| <b>Overall, how satisfied are you with the information in this book?</b> | Satisfied                | Dissatisfied             |
|                                                                          | <input type="checkbox"/> | <input type="checkbox"/> |

|                                                                    |                          |                          |
|--------------------------------------------------------------------|--------------------------|--------------------------|
| <b>How satisfied are you that the information in this book is:</b> | Satisfied                | Dissatisfied             |
| Accurate                                                           | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete                                                           | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to find                                                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to understand                                                 | <input type="checkbox"/> | <input type="checkbox"/> |
| Well organized                                                     | <input type="checkbox"/> | <input type="checkbox"/> |
| Applicable to your task                                            | <input type="checkbox"/> | <input type="checkbox"/> |

Specific comments or problems:

---

---

---

Please tell us how we can improve this book:

---

---

---

Thank you for your comments. If you would like a reply, provide the necessary information below.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_

\_\_\_\_\_  
Phone No.

\_\_\_\_\_



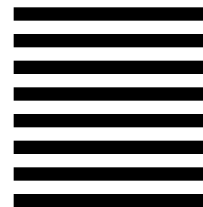
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Design & Information Development  
IBM Corporation  
Software Reengineering  
Department G71A/ Bldg 503  
P.O. Box 12195  
Research Triangle Park, NC 27709-9990



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5688-035



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC30-3470-13

