Network Control Program
System Support Programs
Emulation Program

Resource Definition Reference

IBM

Network Control Program
System Support Programs
Emulation Program

Resource Definition Reference

> **Note**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page ix.

Contents

List of Abbreviations, Glossary, Bibliography, and Index

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them 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 that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described 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
> 500 Columbus Avenue
> Thornwood, NY 10594
> USA

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:

> Site Counsel
> IBM Corporation
> P.O. Box 12195
> 3039 Cornwallis
> Research Triangle Park, NC 277092195
> USA

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

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.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

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

Advanced Peer-to-Peer Networking	IMS	Selectric
	Library Reader	SystemView
AIX	LPDA	System/360
AnyNet	MVS/ESA	System/370
APPN	NetFinity	VM/ESA
BookManager	NetView	Virtual Machine/
CUA	NTune	Enterprise Systems
ESCON	NTuneMON	Architecture
ES/9370	NTuneNCP	VSE/ESA
Hardware Configuration Definition	Nways	VTAM
	OS/2	
IBM	OS/390	

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

This book provides information to help you define resources for the Advanced Communications Function for Network Control Program (NCP) and the Emulation Program (EP). It contains the definition statements and keywords you need to code an NCP generation definition.

Who Should Use This Book

This book is for system analysts, system programmers, system engineers, and field support personnel responsible for defining NCP or EP for an IBM 3745 Communication Controller. In addition to a complete set of specifications for the resources in your network, you need a good understanding of the following:

- Systems Network Architecture (SNA)[1] and the functions NCP provides in an SNA network

- The hardware and software requirements for NCP[2]

- Your network configuration

How to Use This Book

Before you begin to define resources for your NCP, talk with the person responsible for configuring the communication controller in which your NCP will run. Determine the following:

- The physical and operational characteristics of your controller (for example, the number of channel attachments possible and the amount of storage available)

- The specific resources in your network (for example, type 2.1 node support or dynamic reconfiguration)

If you are defining resources for NCP V7R7 for the first time and your generation definition is for an earlier release, you might need to update definition statements or keywords that have changed since your earlier release. You can use the NCP/EP definition facility (NDF) migration aid function to make these changes automatically, or you can make the changes manually. Refer to the *NCP V7R7 Migration Guide* for a description of the migration aid function and a list of the definition statements and keywords that have changed since your earlier NCP release.

At any time after you have updated your generation definition to generate NCP V7R7, you might want to add functions to your NCP. These might be functions added to NCP since your earlier release or functions you have not needed until now. You might also want to change existing definition statements or keywords to change certain characteristics of your NCP. Refer to the *NCP V7R7 Migration Guide* for a brief description of the functions added for each NCP release and the definition statements and keywords added or changed for each function. Refer to the *NCP, SSP, and EP Resource Definition Guide* for a complete description of any NCP function and a list of the definition statements and keywords you need to code

---

[1] Refer to *Systems Network Architecture Technical Overview* for this information.

[2] Refer to *Planning for NetView, NCP, and VTAM* for this information.

for it. The *NCP, SSP, and EP Resource Definition Guide* also contains short examples that show you how an NCP function might be coded. At the end of many chapters, you can find longer examples, and Appendix A contains complete generation definition examples to help you plan and code your own NCP generation definition.

Once you have identified the definition statements and keywords you want to add or change, look them up in Chapter 2, "Definition Statement and Keyword Descriptions" in this book to find out their current syntax and function. Definition statements appear in alphabetical order, and keywords appear in alphabetical order under each definition statement. Keywords that can be coded on more than one definition statement appear under the lowest-level definition statement on which they can be coded.

If you need more information on the conventions used in this book or on coding generation definitions in general, read Chapter 1, "Getting Started." If you already know the function of a keyword and need a quick reminder of its syntax or the definition statements it can be coded on, it might be more convenient to look it up in Appendix A, "Keyword Summary," an alphabetized summary of all the NCP keywords.

Terms Used in This Book

MVS, VM, and VSE
The term *MVS* means the MVS/ESA and OS/390 systems. The term *VM* means the VM/ESA systems in the CMS environment. The term *VSE* means the VSE/ESA operating system. If information applies to only one system, the specific system name is used.

Emulation
The terms *emulation mode* and *EP* generally refer to both forms of emulation: Partitioned Emulation Programing (PEP) and Emulation Program standalone (EPSA). When *emulation mode* is used in an obvious PEP context or in an obvious EPSA context, it refers only to PEP or only to EPSA.

Port and Channel with LPDA
In discussions concerning link problem determination aid (LPDA) for multiport and data multiplex mode (DMPX) modems, the terms *port* and *channel* are synonymous. Although *port* is the more commonly used term, *channel* can be used in sections describing LPDA.

IBM Special Products or User-Written Code
IBM special products include Network Terminal Option (NTO), Network Routing Facility (NRF), and X.25 NCP Packet Switching Interface (NPSI).

NTuneMON and Its Tuning Feature NTuneNCP
NTuneMON is a program that runs with NetView and monitors NCPs that were activated by the VTAM on the host where NTuneMON is running. If you have the NTuneMON tuning feature, NTuneNCP, you can also tune NCP resources.

For more information about these products, refer to the *NTuneMON V2R5 User's Guide* and the *NTuneNCP Feature Reference*.

IBM 3745 Communication Controller Model Numbers

In this book, the term *IBM 3745 Communication Controller* refers to all IBM 3745 models. When particular models are discussed, the appropriate model numbers are specified. Model numbers include IBM 3745-130, 3745-150, 3745-160, 3745-170, 3745-17A, 3745-210, 3745-21A, 3745-310, 3745-31A, 3745-410, 3745-41A, 3745-610, and 3745-61A.

CSS and 3746 Model 900

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

Token Ring

NCP can connect to an IBM Token-Ring Network using the NCP/Token-Ring inter-connection (NTRI) or the 3746 Model 900 connectivity subsystem attachment. This book uses the term *Token Ring* when referring to either type of connection.

Frame Relay

To support frame-relay networks, NCP can use a transmission subsystem (TSS) or high performance transmission subsystem (HPTSS) adapter on the 3745, or NCP can use a communication line processor (CLP) adapter on the 3746 Model 900 connectivity subsystem. In addition, NTRI frame handler connections use type 1 and type 2 Token-Ring interface couplers (TICs). Unless otherwise stated, this book uses the term *frame relay* when referring to a 3745 or a 3746 Model 900 connection.

Integrated Services Digital Network (ISDN)

Integrated services digital network (ISDN) is a digital end-to-end telecommunication network that supports multiple services including, but not limited to, voice and data. ISDNs are used in public and private network architectures. Starting with NCP V7R5, the 3746 Model 900 connectivity subsystem (CSS) supports an interface into existing ISDN environments for SNA sessions routed through the CSS.

RIP (Routing Information Protocol)

RIP-1 means RIP Version 1, RIP-2 means RIP Version 2, and RIP means either RIP-1 or RIP-2.

What Is New in This Book

This edition contains the following new NCP and System Support Programs (SSP) information, as well as editorial and technical changes. New or changed information is identified by a vertical bar (|) in the left margin.

Displaying or Modifying Model 900 Frame-Relay Congestion Parameters

Certain Model 900 frame-relay congestion parameters can now be tuned using NTuneMON. For example, see CRATEDLY, DSCRDDLY, FRSPRI, HLOQSTOP, QSIZE, and QSIZEDE in the Table of LINE Keywords on page 2-172.

Specifying the Number of Frames Queued for a TSS or HPTSS Adapter

The TRANSBUF keyword is added to the LINE statement to enable you to specify the number of frames that can be queued for a TSS or HPTSS adapter awaiting transmission over a 3745 frame-relay physical line. See TRANSBUF on the LINE definition statement on page 2-285.

Remote Access to NCPROUTE in the Absence of SNALINK

In the absence of SNALINK, an alternative, indirect path for remote NCPs to NCPROUTE is available via an NCST connection to a local NCP with an IP channel connection to the NCPROUTE host. See Exception to P2PDEST/HOSTADDR Matching on page 2-158.

Improved Bandwidth Management for 3746 Model 900

Three new functions give additional control over 3746 Model 900 bandwidth.

Committed Information Rate on NCP Controlled 3746 Model 900 Lines

The new CIR keyword controls whether the 3746 Model 900 uses its committed information rate (CIR) algorithms to manage bandwidth on attached frame-relay and ISDN physical lines. See CIR on the LINE definition statement on page 2-206.

Consolidated Link Layer Management on NCP Controlled 3746 Model 900 Lines

The new CLLM keyword controls whether consolidated link layer management (CLLM) will be used on the 3746 Model 900 attached frame-relay physical line. See CLLM on the LINE definition statement on page 2-207.

DLCI Bandwidth Management

The new DLCI statement enables you to control DLCI bandwidth management. See "DLCI Definition Statement" on page 2-83.

Connection Balancing for 3745 Frame-Relay DLCIs

The following keyword has been added to enable connection balancing of frame-relay peripheral Boundary Access Node (BAN) connections over 3745 frame-relay DLCIs. See BALANCE on the DLCI definition statement on page 2-86.

Support for IP over Switched Frame-Relay Physical Lines

IP is now supported over switched frame-relay physical lines. For example, see the paragraph about coding FRELAY=(LOGICAL,IP) on page 2-119.

Support for Additional VTAM Keywords

The VTAMTOPO keyword can be coded on the BUILD, GROUP, LINE, or PU definition statement. For example, see the VTAMTOPO syntax on the BUILD Statement on page 2-9.

Any VTAM keyword can be coded by preceding it with the VTM. prefix. See the paragraph about the VTM. prefix on page 1-1.

Support for 3746 Model 900 Switched Frame-Relay Physical Lines

You can now code DIAL=YES for 3746 Model 900 frame-relay physical lines. For example, see 3746 Model 900 Frame-Relay Physical Lines on page 2-114.

VTAM Support for the TGCONF and MLTGPRI Keywords

For switched PUs, the TGCONF and MLTGPRI keywords can be coded in the VTAM switched major node. See the paragraph added to the TGCONF keyword description on page 2-419 and the paragraph added to the MLTGPRI keyword description on page 2-250.

Availability of Checksum between Subarea Stations

Checksum protection is now available for PIUs transmitted over subarea links. See PIUCHECK on the PU definition statement on page 2-408.

Modem Clocking Allowed for EP

V25BIS=(YES,DLSDLC) can now be coded for EP. See the note about V.25 bis support for emulation mode on page 2-295.

Increase of IPQLIM Keyword Default Value

For 3745 frame-relay IP interfaces, the default value for the IPQLIM keyword on the PU statement is now 3 times the MAXFRAME value on the LINE statement. See IPQLIM on the PU definition statement on page 2-384.

Integration of EP Standalone (EPSA)

Information about defining an EPSA generation is moved into this book from the old *Emulation Program: Resource Definition and Diagnosis* manual, SC31-6205-0. (Additional information is moved from the *Emulation Program* manual into various manuals in the NCP library, and the *Emulation Program* manual is obsolete beginning with NCP V7R7.) This reorganization of information reflects the improvements provided by EP Release 14, including the integration of EPSA with the NCP and EP distribution tapes and the ability of EPSA to run on all models of the 3745.

Where to Find More Information

A good place to start any task regarding NCP, SSP, or EP is in the *NCP V7R7, SSP V4R7, and EP R14 Library Directory*. This directory introduces the enhancements for the current release and shows where these enhancements are described in the NCP library. It gives you an overview of NCP, SSP, and EP and directs you to information on a variety of tasks related to these programs. When you are using the book online, you can use *hypertext links*[3] to move directly from task and enhancement descriptions to the appropriate chapters of other books in the library.

---

[3] A *hypertext link* is a pointer from a location in an online book to another location in the same book or another book. By selecting highlighted information, such as a message number, you can move quickly to related information and, if desired, back again.

Information for NCP Tasks

The books in the NCP, SSP, and EP hardcopy library are listed below according to
task, along with closely related books and tools you might find helpful. See
"Bibliography" on page X-31 for a brief summary of each book in the NCP, SSP,
and EP library and listings of related publications.

Table 0-1. Sources of Information by Task for NCP V7R7			
Order No.	**Title**	**Hardcopy**	**Softcopy**
Planning			
SC31-8063	*Planning for NetView, NCP, and VTAM*		▪
SC31-8062	*Planning for Integrated Networks*		▪
SC30-3971	*NCP V7R7, SSP V4R7, and EP R14 Library Directory*	▪	▪
SC30-3470	*NCP Version 7 and X.25 NPSI Version 3 Planning and Installation*	▪	▪
Installation and Resource Definition			
SC31-6221	*NCP, SSP, and EP Generation and Loading Guide*	▪	▪
SC30-3889	*NCP V7R7 Migration Guide*	▪	▪
SC31-6223	*NCP, SSP, and EP Resource Definition Guide*	▪	▪
SC31-6224	*NCP, SSP, and EP Resource Definition Reference*	▪	▪
Customization			
LY43-0031	*NCP and SSP Customization Guide*		▪
LY43-0032	*NCP and SSP Customization Reference*	▪	▪
Operation			
SC31-6222	*NCP, SSP, and EP Messages and Codes*	▪	▪
Diagnosis			
LY43-0033	*NCP, SSP, and EP Diagnosis Guide*	▪	▪
LY43-0037	*NCP, SSP, and EP Trace Analysis Handbook*	▪	▪
LY43-0029	*NCP and EP Reference*	▪	▪
LY43-0030	*NCP and EP Reference Summary and Data Areas*	▪	▪
LY30-5610	*NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning*	▪	▪
Monitoring and Tuning			
SC31-6266	*NTuneMON V2R5 User's Guide*	▪	▪
LY43-0039	*NTuneNCP Feature Reference*	▪	▪

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 and are 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 softcopy books by using BookManager READ products or
by using the IBM Library Reader product included on CD-ROM. For more informa-
tion on CD-ROMs and softcopy books, see *IBM Online Libraries: Softcopy Col-*

| *lection Kit User's Guide* and BookManager READ documentation. Or see the
| BookManager home page at:

| `http://booksrv2.raleigh.ibm.com`

| World Wide Web

| Visit the NCP Home Page at:

| `http://www.networking.ibm.com/375/375prod.html`

Chapter 1. Getting Started

This chapter describes the conventions used in this book to describe the NCP definition statements and keywords and discusses some general considerations for coding an NCP generation definition. Specifically, it covers the following topics:

- How definition statement and keyword information is presented
- Keyword syntax diagrams
- Keyword defaults and NDF additions
- Generation definition coding conventions
- Definition statement order
- Coding lower level keywords on higher level definition statements
- File names, prefixes, and labels reserved by NCP or SSP
- NCP coding samples

How Definition Statement and Keyword Information Is Presented

Chapter 2 contains detailed information on each NCP definition statement and keyword. There is a separate section for each NCP definition statement; these sections appear in alphabetical order. Each section contains the following subsections.

Usage: This subsection lists the NCP functions or resources for which the definition statement is used.

Position: This subsection tells where the definition statement must appear relative to other definition statements in your generation definition.

Keywords: This subsection contains a table that shows the following for each keyword described under the definition statement:

- The page on which the keyword is described

- Whether the keyword is required or optional when you code the definition statement

- The definition statements from which values can sift down to the keyword (for more information, see "Coding Lower Level Keywords on Higher Level Definition Statements" on page 1-11)

- The default value for the keyword

- Any restrictions on the use of the keyword

This subsection also lists any VTAM keywords that can be coded on the definition statement. These keywords must appear in the generation definition that is the input to the VTAM initialization process, but they are not required in the generation definition you use to generate NCP. The NCP generation procedure does not check these keywords for proper syntax or verify that any related keywords are present or absent. For more information on VTAM keywords, see the *VTAM Resource Definition Reference* manual.

Any VTAM keyword not in the **VTAM Keywords** list under a definition statement in this book can be specified by coding VTM. as a prefix in front of the VTAM keyword.

1-1

Keyword Descriptions: This subsection describes each keyword in detail. The keyword descriptions appear in alphabetical order. Each description includes the following information:

- The syntax of the keyword
- The specific use of the keyword, if any
- The default, minimum, and maximum values of the keyword, if any
- A complete description of the function of the keyword
- Any restrictions or special considerations for using the keyword

Generally, the following conventions apply to the keyword descriptions:

- General information for all uses of the keyword appears at the beginning of the keyword description.

- If a keyword is for a particular function, resource, environment, or piece of hardware, or if the keyword is required, this is stated at the beginning of the keyword description; for example:

  Use: Required for NTRI

  or

  Use: Start-stop or BSC lines in network control mode

- If the keyword applies to EP, the "Use:" line includes either the term "emulation mode" or the term "EP.".

- Default, minimum, and maximum values also appear at the beginning of the keyword description when these values apply to all uses of the keyword. For example:

  Default: 100
  Minimum: 100
  Maximum: 8000

  If these values are different for different uses of the keyword, they are given in the text of the keyword description.

- A boldface phrase highlights information for specific functions; for example:

  3746 Model 900 Token Ring: Code ADAPTER=TIC3 when defining a group of 3746 Model 900 Token-Ring physical lines.

  This type of information usually appears after the description of the general purpose of the keyword.

Keyword Syntax Diagrams

This book uses standard IBM syntax diagrams to describe the syntax of the NCP keywords. These diagrams use the following conventions:

- Read the diagrams from left to right. In general, any path between the start symbol (►►) and the end symbol (►◄) represents valid coding for a keyword.

  ►►──┤ Syntax diagram ├──►◄

- A long diagram may be broken into two or more lines.

  ►►──┤ First line ├──►

```
►──┤ Second line ├──►◄
```
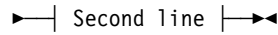
- A keyword that appears on the main path is required. You must code all required keywords.

```
►►──KEYWORD=value──►◄
```
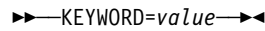
- A keyword that appears below the main path is optional. You do not need to code optional keywords. Most NCP keywords are optional.

```
►►─────────────────►◄
    └─KEYWORD=value─┘
```
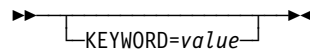
- When you can choose from more than one keyword value, those values are stacked vertically below the main path.

```
►►──────────────────────►◄
    └─KEYWORD=─┬─VALUE1─┬─┘
              └─VALUE2─┘
```
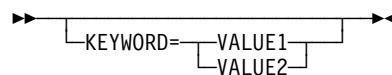
- If a keyword has a default value, the keyword and value appear above the main path.

```
    ┌─KEYWORD=VALUE1───────┐
►►──┤                      ├──►◄
    └─KEYWORD=─┬─VALUE1─┬──┘
              └─VALUE2─┘
```
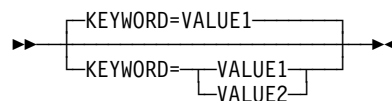
- Uppercase characters show values you code exactly as shown (VALUE1 below). Uppercase characters in a mixed-case string indicate that you can code an abbreviation (VALue2 below means you can code VAL or VALUE2). Lowercase italics show variables for which you need to supply a value, such as a number or string (*value3* below). Do not code a space or comma between the digits of a numeric value.

```
►►──────────────────────►◄
    └─KEYWORD=─┬─VALUE1─┬─┘
              ├─VALue2─┤
              └─value3─┘
```
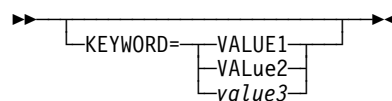
- Multiple keyword values in parentheses are called *suboperands*. Required suboperands appear on the main path between the parentheses (*value1* below). Optional suboperands appear below the main path between the parentheses (*value2* below). Default values for optional suboperands appear above the main path between the parentheses (VALUE2A below). All suboperands must be separated by commas.

```
►►───────────────────────────────────►◄
    └─KEYWORD=(value1─┬─,VALUE2A─┬─)─┘
                      └─,value2──┘
```
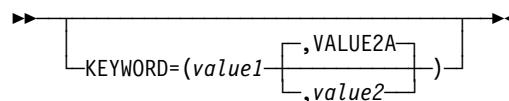
Note: If a default is not shown for an optional suboperand, this means that either there is no default or that the default depends on what you code on other keywords or suboperands. You have to read the keyword description to find out about this.

- An arrow returning to the left above a suboperand indicates that you can code multiple values, enclosed in parentheses and separated by commas.

Keyword Syntax Diagrams

```
▶▶──KEYWORD=(──┬─,◀──────┐──)──▶◀
              └──▼─value──┘
```
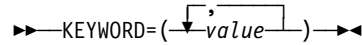
- An arrow returning to the left above a stacked list of suboperands means that you can code as many of the suboperands as you need and in any order. Do not code the same suboperand more than once. Separate each suboperand you code with a comma. For example you could code KEYWORD=(VALUE3,VALUE1) for the keyword below.

```
▶▶──┬─────────────────────────────┬──▶◀
    │            ┌─,◀────────┐     │
    └─KEYWORD=(──▼──┬─VALUE1─┬─┘─)─┘
                    ├─VALUE2─┤
                    └─VALUE3─┘
```
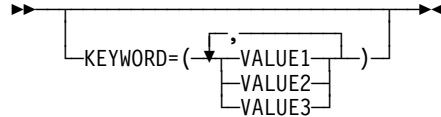
- If you omit an optional suboperand, code a comma to indicate its position. For example, you could code KEYWORD=(VALUE1B,,VALUE3B) for the keyword below to omit the second suboperand. Commas are not required if you omit optional suboperands at the end of a suboperand list. For example you could code KEYWORD=(VALUE1B,VALUE2A) or KEYWORD=(VALUE1B,VALUE2A,) to omit the last suboperand for the keyword below; the result would be the same. If you code only the first suboperand, parentheses are not required. For example you could code KEYWORD=VALUE1B or KEYWORD=(VALUE1B) for the keyword below; the result would be the same. If you omit all suboperands or do not code the keyword, the defaults are used. For example you could code KEYWORD=(), KEYWORD=(,,), or not code the keyword at all for the keyword below; the result would be the same.

```
▶▶──┬───────────────────────────────────────────────────┬──▶◀
    │          ┌─VALUE1A─┐   ┌─VALUE2A─┐   ┌─VALUE3A─┐    │
    └─KEYWORD=(─┼─VALUE1A─┼─,─┼─VALUE2A─┼─,─┼─VALUE3A─┼─)─┘
               └─VALUE1B─┘   └─VALUE2B─┘   └─VALUE3B─┘
```
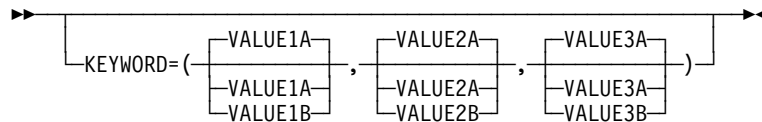
- Restrictions on the use of a keyword value are indicated by superscript numbers in parentheses and are explained below the diagram. Do not code the superscript number or parentheses.

```
▶▶──┬─────────────────────┬──▶◀
    └─KEYWORD=─┬─VALUE1────┬┘
              └─VALUE2─(1)─┘
```
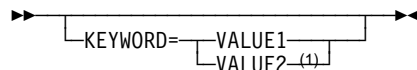
[1] Restriction on VALUE2.

- A section of a long diagram may appear separately below the main diagram.

```
▶▶──┬───────────────────────────────┬──▶◀
    └─KEYWORD=(─┤ Suboperands ├─)─────┘
```

Suboperands:

```
├──┬─VALUE1A─┬──,──┬─VALUE2A─┬──,──┬─VALUE3A─┬───────────────┤
   ├─VALUE1A─┤     ├─VALUE2B─┤     ├─value3──┤
   └─VALUE1B─┘     └─VALUE2C─┘                └─,value4─┘
```
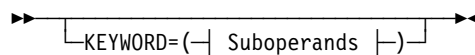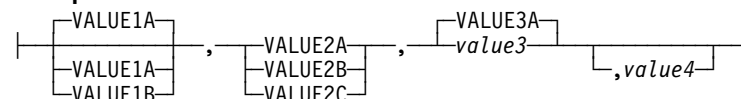
Keyword Defaults and NDF Additions

If you do not code a particular keyword, NDF may still include the keyword in the generation process. This can happen in one of two ways:

- **Default values:** If you do not code a keyword that is used for a resource you are defining and the keyword has a default value, NDF processes your generation definition as if you coded the keyword with the default value. The description of each keyword in this book includes its default value, if any.

- **NDF-added values:** When you are defining certain NCP resources (such as NTRI or frame relay) and you do not code certain keywords or suboperands that are used for those resources, NDF adds those keywords or suboperands to your generation definition with particular values. The NDF-added value for a keyword or suboperand is not necessarily the same as the default. The description of each keyword in this book includes its NDF-added value, if any.

  When your generation definition includes resources whose keywords have NDF-added values, you must code NEWDEFN=YES on the OPTIONS definition statement. This causes NDF to create a *NEWDEFN file*, which contains the definition statements and keywords you coded in the input generation definition and any definition statements and keywords added by NDF. The NEWDEFN file is used during the VTAM initialization process. For more information, see NEWDEFN on the OPTIONS definition statement on page 2-344 in this book or read the discussion of NDF-generated definition files in the *NCP, SSP, and EP Resource Definition Guide*.

The default or NDF-added value for some keywords depends on how you code other keywords. In addition, there may be only one valid value for a keyword depending on how you code other keywords. This book presents these defaults, NDF-added values, and valid values in tables similar to Table 1-1.

Table   1-1. NDF Values for TYPE on the LINE Definition Statement (Sample)

When You Code This on GROUP	The Effect on TYPE Is
ECLTYPE	NDF adds TYPE=NCP. The only valid value is TYPE=NCP.
ETHERNET=PHYSICAL	The default is TYPE=NCP.

The first row shows that if you code the ECLTYPE keyword on the GROUP definition statement, then (1) if you do not code the TYPE keyword on the LINE definition statement, NDF adds TYPE=NCP to the generation definition, and (2) if you code the TYPE keyword, you must code TYPE=NCP. The second row shows that if you code ETHERNET=PHYSICAL on the GROUP definition statement and you do not code the TYPE keyword on the LINE definition statement, NDF uses TYPE=NCP during the generation process.

Generation Definition Coding Conventions

This section describes the conventions for coding definition statements, keywords, and comments in an NCP generation definition. These coding conventions are similar to those for any assembler language program.

Generation Definition Coding Conventions

Definition Statements and Keywords: When coding definition statements and keywords, follow these conventions:

- Use the following general format for coding a definition statement and its keywords:

  ```
  SYMBOL    STATEMENT    KEYWORD1=VALUE1,KEYWORD2=VALUE2,...
  ```

- The **symbol** (also known as a *label*) provides a name for the resource defined by the definition statement. (The symbol is optional for many definition statements.) Code the symbol starting in column 1. The first character of a symbol must not be *$*. Do not code the same symbol for more than one definition statement.

- If a **definition statement** has no symbol, code the definition statement starting in column 10. If a definition statement does have a symbol, code the definition statement starting in column 10 or later on the same line as the symbol.

- Code the **first keyword** of a definition statement on the same line as the definition statement. Code at least one blank space between the definition statement and the keyword.

- To code the **next keyword** on the same line, do the following:

  - Code a comma immediately after the preceding keyword.

  - Code the new keyword immediately after the comma.

- To code the **next keyword** on a new line, do the following:

  - Code a comma immediately after the preceding keyword.

  - Code a nonblank character in column 72 following the preceding keyword.

  - Code the new keyword starting in column 16.

- Do not code a comma after the **final keyword** of a definition statement. A blank space following a keyword indicates the end of the definition statement.

- Do not extend any line beyond column 71.

Comments: When coding comments, follow these conventions:

- You can use a **whole line** as a comment. Code an asterisk in column 1 to indicate that everything on the line is a comment.

- You can code a brief comment on the **same line as a keyword**. Code at least one blank space between the keyword and the comment.

- Do not extend any comment line beyond column 71.

Example: Figure 1-1 on page 1-7 illustrates these coding conventions.

```
1        10   16                                                        72
↓        ↓    ↓                                                         ↓
********************************************************************
*                    DEFINITIONS FOR DEPARTMENT 016              *
********************************************************************
GRP016   GROUP LNCTL=SDLC,                                             X
                TYPE=NCP,                                              X
                DUPLEX=FULL,                                           X
                SPEED=9600,                                            X
                CLOCKNG=EXT,                                           X
                ACTIVTO=420     ** INCREASED TO 420, 12/6/91 **
***
LIN016   LINE ADDRESS=(012,FULL),SDLCST=(SDLPRIB,SDLSECB),            X
                MONLINK=YES,ISTATUS=ACTIVE
***
PU016    PU ISTATUS=INACTIVE,ANS=CONT,PUTYPE=4,TGN=ANY
```

Figure 1-1. Coding Example for Definition Statements, Keywords, and Comments.   Column numbers are highlighted at the top.

Definition Statement Order

Table 1-2 lists the definition statements for various NCP characteristics in the order in which they must appear in your generation definition. For a complete description of the sequence and usage requirements for all the definition statements, see the *NCP, SSP, and EP Resource Definition Guide*.

Table 1-2 does not list the ADD and DELETE definition statements because you code them in a separate generation definition for dynamic reconfiguration. For more information, see the descriptions of these definition statements in Chapter 2.

Table 1-2 (Page 1 of 5). Definition Statement Order

Definition Statement	Required	Notes
GENERATION PROCESS CONTROL DEFINITION STATEMENT		
OPTIONS		Controls network generation processing.
SYSTEM DEFINITION STATEMENTS		
PCCU	Required	VTAM only—identifies the communication controller in which NCP resides.
BUILD	Required	Specifies characteristics of the entire NCP. Begins an NCP generation definition unless you code the OPTIONS definition statement first or code one or more PCCU definition statements.
SYSCNTRL	Required	Specifies dynamic control facilities.
NCPNAU		Defines resources for user-written code and programmed resources.
GWNAU		Defines resources for SNA network interconnection (SNI).

Definition Statement Order

Table 1-2 (Page 2 of 5). Definition Statement Order

Definition Statement	Required	Notes
CONFIGURATION DEFINITION STATEMENTS		
HOST	Required	VTAM only—specifies characteristics of the host processor access method channel-attached to NCP.
IDLIST		Specifies identification sequences for binary synchronous communication (BSC) or teletypewriter exchange service (TWX) devices attached to switched telecommunication links managed by TCAM.
LUPOOL		Defines resources for devices attached to switched telecommunication links.
LUDRPOOL		Defines resources for dynamic reconfiguration or switched SDLC telecommunication links.
PUDRPOOL		Defines resources for dynamic reconfiguration.
PATH	Required	Specifies characteristics of virtual and explicit routes to destination subareas.
SDLCST		Specifies link characteristics for point-to-point subarea links.
DIALSET		Defines a dial set for BSC or start-stop devices attached to switched telecommunication links.
MTALCST		Defines resources for multiple terminal access (MTA) for switched telecommunication links.
MTALIST		Defines resources for MTA for switched telecommunication links.
MTAPOLL		Defines resources for MTA for switched telecommunication links.
MTATABL		Defines resources for MTA for switched telecommunication links.
SWMMT		Defines resources for SSCP monitor mode functions (SMMFs).
NETWORK CONFIGURATION DEFINITION STATEMENTS		
Start-Stop Lines		
GROUP	Required	Specifies link characteristics for point-to-point subarea links.
LINE	Required	Defines a line.
TERMINAL		Defines resources for BSC or start-stop devices.
COMP		Defines resources for BSC or start-stop devices.
ENDAUTO		Indicates the end of an autoline or autocopy definition statement structure.
BSC Lines		
GROUP	Required	Defines a group of lines with similar characteristics.
LINE	Required	Defines a line.
SERVICE		Defines resources for nonswitched telecommunication links.
CLUSTER		Defines resources for clustered BSC stations.
TERMINAL		Defines resources for BSC or start-stop devices.
COMP		Defines resources for BSC or start-stop devices.
ENDAUTO		Indicates the end of an autoline or autocopy definition statement structure.

Table 1-2 (Page 3 of 5). Definition Statement Order

Definition Statement	Required	Notes
SDLC Lines		
GROUP	Required	Defines a group of lines with similar characteristics.
LINE	Required	Defines a line.
SERVICE		Defines resources for nonswitched telecommunication links.
PU		Defines resources for SDLC devices.
LU		Defines resources for nonswitched telecommunication links.
ENDAUTO		Indicates the end of an autoline or autocopy definition statement structure.
Stand-Alone SDLC Lines		
GROUP	Required	Defines a group of lines with similar characteristics.
Token-Ring Lines		
GROUP	Required	Defines a group of Token-Ring logical or physical lines.
LINE	Required	Defines a line.
PU		Defines resources for Token-Ring devices.
IPHOST		Defines resources for an internet router.
Frame-Relay Lines		
GROUP	Required	Defines a group of frame-relay logical or physical lines.
LINE	Required	Defines a line.
DLCI		Identifies the DLCIs for which the statement's keywords apply.
PU		Defines resources for frame-relay devices.
ISDN Lines		
GROUP	Required	Defines a group of ISDN logical or physical lines.
LINE	Required	Defines a line.
PU		Defines resources for ISDN devices.
Ethernet-Type LAN Lines		
GROUP	Required	Defines a group of Ethernet-type LAN physical lines.
LINE	Required	Defines a line.
PU		Defines resources for Ethernet-type LAN devices.
IPHOST		Defines resources for an internet router.

Definition Statement Order

Table 1-2 (Page 4 of 5). Definition Statement Order

Definition Statement	Required	Notes
NCP Connectionless SNA Transport (NCST) Sessions		
GROUP	Required	Defines a group of NCST logical lines.
LINE	Required	Defines a line.
PU	Required	Defines resources for NCST sessions.
LU	Required	Defines NCST session interfaces.
IPLOCAL	Required	Defines NCST session interface characteristics.
Channel Links		
GROUP	Required	Defines a group of lines with similar characteristics.
LINE	Required	Defines a line.
PU		Defines resources for channel devices.
LU		Defines resources for nonswitched telecommunication links.
User Lines		
GROUP	Required	Defines a group of lines with similar characteristics.
LINE	Required	Defines a line.
PU		Defines resources for channel link stations.
LU		Defines resources for type 2.1 channel link stations.
ENDAUTO		Indicates the end of an autoline or autocopy definition statement structure.
INTERNET RESOURCE DEFINITION STATEMENTS		
IPOWNER		Identifies the IBM TCP/IP owner of an NCP routing table for MVS.
IPLOCAL		Defines resources for an internet router.
IPROUTE		Defines resources for an internet router.
FRAME-RELAY SWITCHING RESOURCE DEFINITION STATEMENT		
FRSESET		Defines a frame-relay frame handler subport set for frame-relay resources.
SNI NON-NATIVE NETWORK RESOURCE DEFINITION STATEMENTS		
NETWORK		Defines resources for SNI.
GWNAU		Defines resources for SNI.
PATH		Defines resources for SNI.

Table 1-2 (Page 5 of 5). Definition Statement Order

Definition Statement	Required	Notes
BLOCK HANDLER DEFINITION STATEMENTS		
STARTBH		Defines resources for block handling routines.
EDIT		Defines resources for block handling routines.
DATETIME		Defines resources for block handling routines.
UBHR		Defines resources for block handling routines.
REMOVCTL		Defines resources for block handling routines.
ENDBH		Defines resources for block handling routines.
BHSET		Defines resources for block handling routines.
GENERATION DELIMITER DEFINITION STATEMENT		
GENEND	Required	Ends an NCP generation definition unless you code the OPTIONS definition statement last.

Coding Lower Level Keywords on Higher Level Definition Statements

The NCP definition statements fall into a hierarchical order. At the top of the hierarchy are definition statements that define characteristics of the entire network, such as OPTIONS and BUILD. Lower in the hierarchy are definition statements that define resources for large parts of the network, such as GROUP and LINE. At the bottom are definition statements that define individual components of the network, such as TERMINAL and COMP. In general, higher level definition statements appear before lower level definition statements in the generation definition.

For a subset of this hierarchy (GROUP, LINE, PU, LU, CLUSTER, TERMINAL, and COMP, in descending order), some keywords for lower level definition statements can be coded on higher level definition statements. When you do this, the keyword value "sifts" down to the lower level definition statements coded under higher level definition statement. For example, you can code the SPEED keyword on a LINE definition statement to define the speed for that line, or you can code the SPEED keyword on a GROUP definition statement to define the same speed for all the lines in the group. This can reduce the amount of coding in your generation definition. For example, you can specify the same dial set for 15 terminals on a line by coding the DIALSET keyword just once on the LINE definition statement instead of on each of the 15 TERMINAL definition statements.

You can code a keyword on a lower level definition statement to override the value that sifts down from a higher level definition statement. For example, if you want 12 of the 15 terminals on a line to have the same dial set, you can code the DIALSET keyword on the LINE definition statement to define the same dial set for all 15 terminals, and then code the DIALSET keyword on three of the TERMINAL definition statements to define a different dial set for those three terminals. For a keyword with multiple suboperands, *all* the suboperands of the keyword on the higher level definition statement are overridden even if you do not specify all the suboperands on the lower level definition statement. NCP uses the default values for any suboperands you do not specify.

Coding Lower Level Keywords

Table 1-3 lists all the keywords you can code on these definition statements, shows which definition statements you can code them on, and shows you where to find information about the keyword. This table does not indicate the conditions under which each keyword can be used (for example, type of line control or type of device). For this information, see the keyword descriptions in Chapter 2.

Note: The values of certain keywords on the BUILD definition statement sift down to the NETWORK definition statement. For a list of these keywords, see Table 2-97 on page 2-331.

Table 1-3 (Page 1 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

		Definition Statement								
		For Network Control Mode							For Emulation Mode	
Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
ACTIVTO	2-105	X								
ADAPTER	2-106	X	X							
ADDIFG	2-180	•	X							
ADDR	2-366			X			X	•		
ADDRESS	2-180		X							X
ANS	2-193	•	X	X						
ANSTONE	2-193	•	X							
ANSWER		V	V							
ARPTAB	2-371		X	X						
ASLENT		V	V	V	V	V	V			
ASLTAB		V	V	V	V	V	V			
ATTN	2-452	•	•				X	•		
AUTO	2-194		X							X
AUTOCOPY	2-196		X							X
AUTODL	2-198	•	X							
AUTOGEN	2-107	X								
AUTOLINE	2-199		X							X
AUTUACB	2-200	•	X							
AVGPB	2-200	•	X	X						
BALANCE	2-200	•	X							
BEACTO	2-201	•	X							
BFRDLAY	2-452	•	•			•	X			
BHEXEC	2-453	•	•			X	X	•		
BHSET	2-454	•	•			X	X	•		
BLOCK	2-373			X						
BNNSUP	2-374	•	•	X						
BRKCON	2-374	•	•	X						
BUFETTE	2-201								•	X
BUFSIZE	2-201								•	X

X indicates the definition statements under which the keyword is described in this book.
• indicates other definition statements on which the keyword may be coded.
V indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Table 1-3 (Page 2 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

Keyword	Page	| Definition Statement |							For Emulation Mode	
		| For Network Control Mode |								
		GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
CA	2-202	•	X						•	X
CAEXIT	2-107	X								
CALINE	2-202	•	X							
CALL	2-203	•	X							
CAPACITY		V	V							
CASDL	2-203	•	X	X						
CCLID	2-204	•	X							
CHANLA	2-204		X							
CHANLNK	2-108	X								
CHAREC	2-108	X							X	
CHECK	2-205									X
CHNLZ	2-205	•	X							
CHNPRI	2-206									X
CIR	2-206	•	X							
CLINES	2-207		X							
CLOCKNG	2-208	•	X						•	X
CODE	2-210	•	X						•	X
COMPACB	2-109	X								
COMPOWN	2-110	X								
COMPSWP	2-111	X								
COMPTAD	2-112	X								
COMRATE	2-375	•	•	X						
CONFIG	2-212	•	X							
CONNTYPE		V	V							
CONV	2-454	•	•				X	•		
CORNUM	2-213	•	X							
COSBYTE		V	V							
COSTTIME		V	V							
CPCP		V	V							
CRATEDLY	2-214	•	X							
CRDLAY	2-455	•	•				X			
CRETRY	2-112	X								
CRITSIT	2-455	•	•			•	X			
CRRATE	2-214		X							
CSPMODE	2-215								•	X
CTERM	2-456						X			
CU	2-215								•	X

X indicates the definition statements under which the keyword is described in this book.

• indicates other definition statements on which the keyword may be coded.

V indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Coding Lower Level Keywords

Table 1-3 (Page 3 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

| | | Definition Statement | | | | | | | For Emulation Mode | |
| | | For Network Control Mode | | | | | | | | |
Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
CUIDLEN	2-457	•	•				X			
CUTOFF	2-215	•	X							
CUTYPE	2-216	•	X			X			•	X
DATABLK	2-217	•	X							
DATMODE	2-375	•	•	X						
DATRATE	2-217	•	X						•	X
DELAY	2-217	•	X	X					X	
DIAL	2-113	X							X	
DIALALT	2-218	•	X							
DIALNO	2-458						X			
DIALSET	2-218	•	X				X			
DIRECTN	2-460	•	•			•	X			
DISABLE	2-218								•	X
DISCNT		V	V	V		V				
DLCI	2-377			X						
DLOGMOD		V	V	V	V	V	V			
DSCRDDLY	2-219	•	X							
DSRTO	2-114								X	
DUALCOM	2-219									X
DUPLEX	2-220	•	X						•	X
DYNADJCP		V	V							
DYNADMP	2-221								•	X
DYNLU		V	V	V						
DYNPU		V	V							
DYNPUPFX		V	V							
DYNWIND	2-222	•	X	X						
ECLTYPE	2-115	X								
ENCR			V	V	V					
ENDTRNS	2-460	•	•				X	•		
EOB	2-116								X	
EOT	2-116								X	
ERRORT	2-380	•	•	X						
ETHERNET	2-117	X								
ETRATIO	2-224	•	X							
EXEC	2-460	•	•			X	X	•		
FANOUT	2-461						X			
FEATURE	2-461	•	X			X	X		•	X

X indicates the definition statements under which the keyword is described in this book.
• indicates other definition statements on which the keyword may be coded.
V indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Table 1-3 (Page 4 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE

Definition Statement										
		For Network Control Mode							For Emulation Mode	
Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
FEATUR2		V	V	V	V	V	V			
FRAMECNT	2-226		X							
FRAMING	2-117	X								
FRELAY	2-118	X								
FRRATES	2-227	•	X							
FRSPRI	2-381	•	X	X						
GPOLL	2-69					X				
GP3174	2-381	•	•	X						
HDXSP	2-228	•	X							
HICHAN	2-228								•	X
HISPEED	2-229	•	X							
HLOQSTOP	2-229	•	X							
HPR		V	V	V						
HOSTLINK	2-117		X							
HPRQLIM	2-382	•	•	X						
ID	2-230	•	X							
IDSEQ	2-463	•	•				X			
INBFRS	2-230	•	X							
INCLUD0E		V	V	V						
INHIBIT	2-464	•	•			X	X	•		
INNPORT	2-382			V						
INTFACE	2-231		X	X	X					
IPL	2-232	•	X							
IRETRY	2-384	•	•	X						
ISDN	2-119	X								
ISDNCALL	2-232	•	X							
ISDNCFG	2-385	•	•	X						
ISDNDSCT	2-385	•	•	X						
ISDNLPN	2-233	•	X							
ISDNLSA	2-234		X							
ISDNRPN	2-235	•	X							
ISDNRSA	2-235		X							
ISTATUS		V	V	V	V	V	V			
ITBMODE	2-465	•	•			X	X	•		
KBDLOCK	2-120	X								
LADDR	2-156			X						
LANTYPE	2-386	•	•	X						

X   indicates the definition statements under which the keyword is described in this book.
•   indicates other definition statements on which the keyword may be coded.
V   indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Coding Lower Level Keywords

Table 1-3 (Page 5 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

Keyword	Page	For Network Control Mode							For Emulation Mode	
		GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
LCST	2-465	•	•				X			
LEVEL2	2-120	X								
LEVEL3	2-121	X								
LEVEL5	2-122	X								
LGRAPHS	2-466	•	•			X	X	•		
LIMRES	2-236	•	X	V						
LINEADD	2-123	X								
LINEAUT	2-123	X								
LINECB	2-236	•	X							
LINEFVT	2-236	•	X							
LINESIZ	2-237	•	X							
LLERP	2-387	•	•	X						
LMI	2-388	•	•	X						
LMODADR	2-389			X		X	X			
LNCTL	2-124	X							X	
LNKOWNER	2-125	X								
LNQTCNT	2-237	•	X							
LOCADD	2-238		X							
LOCADDR	2-300				X					
LOCALTO	2-239	•	X	•						
LOCHAN	2-240								•	X
LOGAPPL		V	V	V	V	V	V			
LOGTAB		V	V	V	V	V	V			
LPDA	2-390			X						
LPDATS	2-240	•	X							
LPDA2DL	2-243	•	X							
LSPRI	2-244	X	X							
LUCB	2-301	•	•	•	X					
LUDR	2-302	•	•	•	X					
LUFVT	2-302	•	•	•	X					
LUGROUP				V						
LUNTFY	2-303	•	•	•	X					
LUSEED				V						
MAXBFRU	2-391	•	•	X					•	•
MAXCOLL	2-303	•	•	•	X					
MAXDATA	2-391	•	•	X						
MAXDLCI	2-245	•	X	X						

X indicates the definition statements under which the keyword is described in this book.
• indicates other definition statements on which the keyword may be coded.
V indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Table 1-3 (Page 6 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
									For Emulation Mode	
		For Network Control Mode								
MAXFRAME	2-246	•	X							
MAXLU	2-394	•	•	X						
MAXOUT	2-395	•	•	X						
MAXPU	2-248	•	X							
MAXSESS	2-304	•	•	•	X					
MAXTP	2-304	•	•	•	X					
MAXTSL	2-249	•	X							
MDLENT		V	V	V	V	V	V			
MDLTAB		V	V	V	V	V	V			
METRIC	2-157			X						
MLTGPRI	2-250	•	X							
MODE	2-127	X								
MODEM	2-250								•	X
MODETAB		V	V	V	V	V	V			
MODULO	2-399	•	•	X						
MONITOR	2-251	•	X							
MONLINK	2-252	•	X	X						
MPTALT	2-253	•	X							
MTALIST	2-254	•	X							
MULTI	2-254								•	X
NATIVE		V								
NCPCA	2-254	•	X							
NCST	2-127	X								
NEGPOLP	2-255	•	X							
NETID	2-127	X	•	X	X					
NETWORK	2-401			X						
NEWSYNC	2-255	•	X						•	X
NEXT	2-401			X	X	X	X	•		
NN		V	V							
NPACOLL	2-256	•	X	X	X	X	X			
NPARSC	2-128	X								
NPATP	2-257	•	X	X						
NRZI	2-258	•	X							
NUMSESS	2-307	•	•	•	X					
OWNER		V	V	V						
PACING	2-307	•	•	•	X	V				
PAD	2-259								•	X

X   indicates the definition statements under which the keyword is described in this book.
•   indicates other definition statements on which the keyword may be coded.
V   indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Coding Lower Level Keywords

Table 1-3 (Page 7 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

| | | Definition Statement | | | | | | | For Emulation Mode | |
| | | For Network Control Mode | | | | | | | | |
Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
PADCNT	2-128	X								
PARCHK	2-259	•	X							
PARGEN	2-259	•	X							
PASSLIM	2-404	•	•	X						
PAUSE	2-260	•	X							
PDELAY		V	V							
PECHAR	2-303	X								
PHYPORT	2-129	X								
PHYSRSC	2-130	X								
PIUCHECK	2-408			X						
POLIMIT	2-261	•	X							
POLL	2-468						X	•		
POLLED	2-262	•	X							
POLLTO	2-263	•	X							
PORT	2-263		X							
PORTADD	2-263		X							
PRODID	2-131	X								
PROMPT	2-265	•	X							
PROTOCOL	2-158			X						
PT3EXEC	2-469	•	•			X	X	•		
PU		V	V							
PUCB	2-408	•	•	X						
PUDR	2-408	•	•	X						
PUFVT	2-410	•	•	X						
PUNTFY	2-410	•	•	X						
PUTYPE	2-131	X	•	X						
P2PDEST	2-412			X						
QUIET	2-265								•	X
QSIZE	2-265	•	X							
QSIZEDE	2-266	•	X							
QUIETCT	2-132								X	
RCVBUFC	2-266	•	X							
RCVCHAR	2-267								•	X
REDIAL	2-267	•	X							
REGISTER		V	V	V	V					
REMLU	2-308				X					
REMOTTO	2-268	•	X	•						

X indicates the definition statements under which the keyword is described in this book.
• indicates other definition statements on which the keyword may be coded.
V indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Table 1-3 (Page 8 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

| | | Definition Statement | | | | | | | For Emulation Mode | |
| | | For Network Control Mode | | | | | | | | |
Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
REPLYTO	2-132								X	
RESSCB	2-309				X					
RETRIES	2-269	•	X	X						
RETRYTO	2-133	X								
RING	2-272	•	X						•	X
RNRLIMT	2-134	X								
ROLE	2-414	•	•	X						
SDLCST	2-273	•	X							
SECNET				V	V					
SECURE	2-273	•	X							
SECURITY		V	V							
SEGMENT	2-415			X		X	X			
SERVLIM	2-274	•	X							
SERVPRI	2-275	•	X							
SESSION	2-275	•	X							
SESSREQ	2-469						X			
SHOLD		V								
SNETMASK	2-415			X						
SPAN		V	V	V	V					
SPDSEL	2-276	•	X				V			
SPEED	2-276	•	X						•	X
SPOLL	2-416	•	•	X						
SPSHIFT	2-279	•	X							
SRETRY	2-280	•	X							
SRT	2-416	•	•	X			X	•		
SSAP	2-418			X						
SSCPFM		V	V	V	V	V	V			
SUBAREA				V						
SWMMTID	2-280	•	X							
TADDR	2-280	•	X						•	X
TAILING	2-282	•	X							
TERM	2-282	•	X	V	V	•	X		•	X
TEXTTO	2-134	X							X	
TGCONF	2-418	•	•	X						
TGN	2-419	•	•	X						
TGP		V	V							
TIMEOUT	2-284	•	X	X						

X   indicates the definition statements under which the keyword is described in this book.
•   indicates other definition statements on which the keyword may be coded.
V   indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Coding Lower Level Keywords

Table 1-3 (Page 9 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
		For Network Control Mode							**For Emulation Mode**	
TIMER	2-135	X								
TIMERS	2-421	•	•	X						
TITIMER	2-285		X							
TOPO		V	V							
TRANSBUF	2-285		X							
TRANSFR	2-286	•	X	X						
TRSPEED	2-288	•	X							
TTDCNT	2-136	X								
TYPE	2-288	X	X						X	X
T1TIMER	2-423	•	X	X						
T2TIMER	2-291	•	X	X						
UACB	2-292		X							
UCCB	2-309				X					
UNIQUE	2-426		X							
UNITXC	2-293								•	X
UPARM1		V	V							
UPARM2		V	V							
UPARM3		V	V							
USE	2-293	•	X						•	X
USERID	2-137	X								
USSTAB		V	V	V	V	V	V			
VERSALID		V	V							
VIROWNER	2-138	X								
VIRTUAL	2-139	X								
VNGROUP		V	V							
VNNAME		V	V							
VPACING	2-309	•	•	•	X	V				
VPRINT	2-472	•	•			•	X			
VTAMTOPO		V	V	V						
V25BIS	2-294	•	X						•	X
WACKCNT	2-139	X								
WAKDLAY	2-140	X								
WTTYEOB	2-140	X								
WTTYEOT	2-140	X								
XID	2-426	•	•	X						
XIO	2-141	X								
XMITDLY	2-142	X								

X   indicates the definition statements under which the keyword is described in this book.
•   indicates other definition statements on which the keyword may be coded.
V   indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

Table 1-3 (Page 10 of 10). Coding Lower Level Keywords on Higher Level Definition Statements

		Definition Statement							
		For Network Control Mode						For Emulation Mode	
Keyword	Page	GROUP	LINE	PU	LU	CLUSTER	TER-MINAL	COMP	GROUP	LINE
XMONLNK	2-427	•	•	X						
XTWXID	2-473	•	•				X			
X21NTWK	2-142	X								
X21SW	2-143	X								
YIELD	2-295	•	X							

X   indicates the definition statements under which the keyword is described in this book.

•   indicates other definition statements on which the keyword may be coded.

V   indicates the definition statements on which the keyword may be coded for VTAM only. See the *VTAM Resource Definition Reference* for a description of the keyword.

File Names, Prefixes, and Labels Reserved for NCP or SSP

NDF might add certain definition statement labels to your generation definition. Do not code these labels in your generation definition. If you do, NDF processing might create duplicate labels, which can cause assembler errors.

Table 1-4 lists prefixes that NDF uses for definition statement labels. Do not code labels that begin with these prefixes.

Table 1-4. Label Prefixes to Avoid

@	BOQ	CPT	EPI	IX	LTV	NPB	QCB	SNP	UNA
$	BPB	CRB	EQB	Jn	LTXn	NPF	RAT	SOT	USC
AAB	BSB	CRP	ERB	LAA	LU	NQB	RCB	SPC	UXRn
ABN	BST	CTB	ERX	LAB	LX	NQE	RCQ	SST	U1
ACB	BTT	CTP	FCT	LBn	L1B	NSQ	RCV	STE	VAT
ACT	BTU	CUB	FLB	LCB	L4B	NVT	RG	STQ	VIT
ACU	BUE	CY	FMT	LCC	MBF	NVX	RHn	SUT	VLB
AEB	CAn	CX	FVT	LCI	MBX	OLL	RNn	SVT	VR
ALE	CAB	DAE	GCB	LCP	MCT	OLT	RUn	SXB	VST
AST	CAI	DDB	GPT	LCS	MDR	PAB	Rn	SYS	VTS
ATB	CAR	DIA	GRW	LCW	MIB	PAD	RMB	TCB	VVT
ATP	CAT	DPT	GVT	LDAn	MIC	PCB	ROSHn	TET	WCB
ATT	CB	DQB	HWE	LDIn	MIF	PIU	RST	TGB	WRP
AVn	CBB	DRS	HWX	LGT	MIH	PLn	RTR	THn	WU
AXB	CDS	DRX	IB	LKB	MIM	PL2	RVT	TIM	X
BC	CER	DSP	ICE	LKC	MLT	PMF	SCB	TND	XDA
BCU	CGP	DTG	ICI	LNB	MMV	PRB	SEB	TQB	XDB
BER	CHC	DVB	ICW	LNV	MSC	PSA	SGE	TRT	XDH
BGS	CHV	DVI	IDD	LPB	MTF	PSB	SGT	TVS	XID
BH	CIE	DVQ	IDE	LRB	NET	PSI	SHB	UAC	
BHD	CM	ECB	IDL	LRC	NIB	PSP	SID	UAD	
BHR	COE	ECD	IDB	LTC	NIX	PST	SIT	UIB	
BHS	CPI	ECL	IRN	LTR	NLB	PUV	SMB	UIC	
BLU	CPN	EML	IRQ	LTS	NLX	QAB	SMM	ULVSGN	

n indicates a number from 0 to 9.

Table 1-5 on page 1-22 lists names that NDF uses as definition statement labels. Do not code these names as labels.

Table 1-5. Labels to Avoid

ACITRAP	CSPQH2	NCPHIST1	SVCO	THFOM	THTYP1
CAACER	CSPQOFF	NCPLVL	SVCQUT	THLAST	THTYP2
CACCER	CSPQON	NEWLNE	SWQTMQ1	THLOB	THTYP3
CADCER	DCTABND	OLDLNE	SWQTMQ2	THLOM	TMRF
CAECER	DCTSAVEK	PEPQSCNB	TABEND	THMID	TTCUR
CAFCER	D*n*RCB	PEPQSCNM	TABSTAR	THMPF	TTEND
CCPH1	EPLVL	PSCA	THAFIB	THODAIB	TTRECNTR
CCPSAVE	FILLB	ROSSVADDR	THAFIM	THODAIM	TTSKPCNT
CHANSNS1	FILLC	ROSSVCCR	THBCUVVT	THONLY	TTSTAR
CHANSNS2	HDRNENT	ROSSVCCU	THFID	THPSIB	UIHRCCW
CHSVBKSV	ICNTABL1	ROSSWK1	THFIRST	THPSIM	USTAGETR
CHSVH1	LCDBSCB	SECNTRI	THFOB	THTYPO	UTILSTSZ
CSPQH1	LCDSSBIT				

n indicates a number from 0 to 9.

NCP Coding Samples

The *NCP, SSP, and EP Resource Definition Guide* contains a complete NCP generation definition for a sample network that includes a wide variety of NCP functions. It also contains complete system generation coding samples and smaller coding examples that show how to define individual NCP resources. You may find these samples helpful in identifying which definition statements and keywords you need to code for particular NCP functions and in determining what values to code for particular keywords.

Chapter 2. Definition Statement and Keyword Descriptions

This chapter provides a detailed description of each NCP definition statement and keyword. For a description of how definition statement and keyword information is presented or for general information on coding an NCP generation definition, see Chapter 1. For information on specific NCP functions and how to define them, see the *NCP, SSP, and EP Resource Definition Guide*.

ADD Definition Statement

Usage: For dynamic reconfiguration. One for each physical unit you want to add to an existing link or one for each logical unit you want to add to a physical unit.

Position: In a separate generation definition. If you specify a link, the ADD definition statement must precede a PU definition statement. If you specify a physical unit, ADD must precede an LU definition statement.

Keywords: Table 2-1 shows the keyword described under this definition statement.

Table 2-1. Keywords on the ADD Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
TO	2-3	Required			

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions
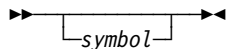
symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

TO

►►—TO=*name*—►◄

Use: Required when you code the ADD definition statement
Default: None

If you want to add a physical unit dynamically, TO specifies the name of the link defined in the generation definition to which the new physical unit will be added. If you want to add a logical unit dynamically, TO specifies the name of the physical unit defined in the generation definition to which the new logical unit will be added.

If you want to add a physical unit to a link, you must code the PU definition statement as described in this book. However, the DATMODE, MAXLU, and SUBAREA keywords are ignored for dynamic reconfiguration. If you omit the RETRIES keyword, the default is 7 for both the *t* and *n* suboperands.

If you want to add a logical unit to a physical unit, you must code the LU definition statement as described in this book. For an example of how to code the ADD definition statement for dynamic reconfiguration, see the *NCP, SSP, and EP Resource Definition Guide*.

BHSET Definition Statement

Usage: BSC or start-stop devices. One for each block handler set needed.

Position: Must follow all block handler definition statements of a block handler set.

Keywords: Table 2-2 shows the keywords described under this definition statement.

Table 2-2. Keywords on the BHSET Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
EXEC	2-5			YES	
PT1	2-5				
PT2	2-5				
PT3	2-5				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the BHSET definition statement
Default: None

Provides a name for the set of block handlers. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates. *symbol* is referred to by the BHSET keyword on the CLUSTER, TER-MINAL, and COMP definition statements.

EXEC

```
            ┌─EXEC=YES─────┐
►►──────────┴──────────────┴──►◄
            └─EXEC=─┬─YES─┐
                    └─NO──┘
```
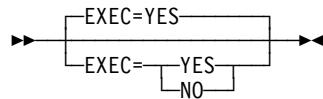
Default: YES

Specifies whether the block handler set can be run when it is associated with a device by command from the host processor.

PT1
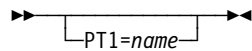
```
►►──┬──────────────┬──►◄
    └─PT1=name─────┘
```

Default: None

Specifies the name of the block handler that is to run when a request for the device is received from the host processor before the line is available.

PT2
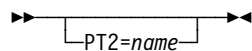
```
►►──┬──────────────┬──►◄
    └─PT2=name─────┘
```

Default: None

Specifies the name of the block handler that is to run when a request for the device is received from the host processor or after receiving data from the device while the line is available.
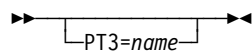
PT3

```
►►──┬──────────────┬──►◄
    └─PT3=name─────┘
```

Default: None

Specifies the name of the block handler that is to run when data is received from the device when the line is no longer available.

BUILD Definition Statement

Usage: Required.

Position: First in the generation definition unless you specify the OPTIONS definition statement first or if you have one or more PCCU definition statements.

Keywords: Table 2-3 shows the keywords described under this definition statement.

*Table 2-3 (Page 1 of 3). Keywords on the BUILD Definition Statement*

Keyword	Page	Required	Sift	Default	Tuning*
ADDSESS	2-10			0	
AUXADDR	2-10			0	
BACKUP	2-11			0	
BFRS	2-11			(240,100)	
BRANCH	2-12			100	
CAIOTRC	2-13			NO	
CATRACE	2-13			(YES,50)	
CNLSQMAX	2-13			10000	
CNLSQTIM	2-14			10	
CSMHDR	2-14				
CSMHDRC	2-14				
CSMSG	2-15				
CSMSGC	2-15				
CSSTIER	2-15				
CUID	2-16				
CWALL	2-17			26	
DIALTO	2-17			60.0	
DR3270	2-17				
DSABLTO	2-18			3.0	
DUPADDR	2-18			WARNING	
DYNNSC	2-18				
DYNNSX	2-18				
DYNPOOL	2-19			(0,0)	Yes*
ENABLTO	2-20			2.2	
ERLIMIT	2-20			8	
GENILU	2-21			YES	
GENLEVEL	2-21				
GWAEXIT	2-22				
GWPACING	2-22			(0,ADAP,ALLOW)	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-3 (Page 2 of 3). Keywords on the BUILD Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
GWSESAC	2-23			NO	
HPR	2-25			YES	
HPRATT	2-26			12000	Yes*
HPRMLC	2-27			9	Yes*
HPRMPS	2-28			0	Yes*
HPRQLIM	2-30			0	
HPRSNAP	2-30			NONE	Yes*
HSBPOOL	2-30				
IPPOOL	2-31				Yes*
IPRATE	2-32				
IPSNAP	2-33			NONE	Yes*
ISDNACC	2-34			YES	
ITEXTTO	2-34			NONE	
LENAME	2-34			INLINKED	
LINETRC	2-35			(YES,4,4)	
LOCALTO	2-35				
LTRACE	2-35			2	
MAXSESS	2-36			65535	
MAXSSCP	2-36				
MAXSUBA	2-37				
MEMSIZE	2-37				
MLTGORDR	2-38			FIFO	
MODEL	2-39	Required			
MTARTO	2-39			1.0	
MTARTRY	2-39			0	
NAMTAB	2-40				
NCPTRACE	2-40			OFF	Yes*
NETID	2-41	Required			
NETLIM	2-41				
NEWNAME	2-41				
NPA	2-42				
NTUNECOL	2-43			NO	Yes*
NUMHSAS	2-44			128	
OLT	2-45			YES	
OWNERPL	2-45			NONE	
PATHEXT	2-46				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table   2-3 (Page 3 of 3). Keywords on the BUILD Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
PRTGEN	2-46			GEN	
PUNAME	2-46				
PWROFF	2-47			NO	
QSZALERT	2-47				Yes*
REMOTTO	2-48			2.5	
RGENLEVL	2-48				
SALIMIT	2-48			255	
SESSACC	2-49			NO	
SESSLIM	2-51			255	
SLODOWN	2-51			12	
SUBAREA	2-52	Required			
TEST	2-52			NO	
TGBXTRA	2-52				
TRACE	2-53			NO	
TRANSFR	2-53				
TRCPIU	2-54			2000	
TWXID	2-54			(XON,XOFF)	
TYPGEN	2-55	Required			
TYPSYS	2-55				
T1TIMER	2-56			2.5 for *remott1*	
T2TIMER	2-56			(0,0,2)	
UCHAN	2-58			NO	
USGTIER	2-59	Required			
VERSION	2-61	Required			
VOSDEACT	2-61			NO	
VRPOOL	2-61				
VRTIMERn	2-62			(0,0,0)	
VSUBNETS	2-64			NO	
XBREAK	2-64			3	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
NTuneMON V2R5 User's Guide for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements: The
values of certain keywords on the BUILD definition statement can sift down to the
NETWORK definition statement. These keywords are indicated in Table 2-97 on
page 2-331. If you code one of these keywords on the BUILD definition statement
but not on the NETWORK definition statement, the value for the BUILD definition
statement also applies to the NETWORK definition statement. If you do not code
the keyword on either definition statement, NCP uses the default value, if any, for
both definition statements.

VTAM Keywords: The following diagrams show the VTAM keywords that can be coded on this definition statement. These keywords provide information to the VTAM initialization process. They are not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for descriptions of these keywords and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

| VTAM keywords not listed below can be specified by coding VTM. as a prefix in
| front of the VTAM keyword.

```
►►─┬──────────────────────────────────────┬─►◄
   └─COSTAB=class-of-service_table_name──┘
```

```
►►─┬──────────────────────────────────────┬─►◄
   └─LOADLIB=ddname_of_NCP_load_module──┘
```

```
►►─┬───────────────────────────────────┬─►◄
   └─QUALIFY=loadlib_name_qualifier──┘
```

```
   ┌─SHM=NO─────────┐
►►─┴────────────────┴─►◄
   └─SHM=─┬─NO──┬─┘
         └─YES─┘
```

```
|  ►►─VTAMTOPO=─┬─INCLUDE──┬─►◄
|              ├─IGNORE───┤
|              ├─NOREPORT─┤
|              ├─REPORT───┤
|              ├─NOSWPUS──┤
|              └─NOLLINES─┘
```

NTuneMON Keywords: The following diagrams show the NTuneMON keywords that can be coded on this definition statement. They are required if you have installed the NTuneMON tuning feature, NTuneNCP, in your communication controller. See the *NTuneMON V2R5 User's Guide* for descriptions of these keywords.

```
►►─ATF.USGTIER=─┬─1───┬─►◄
               ├─2───┤
               ├─2.5─┤
               ├─3───┤
               ├─4───┤
               └─5───┘
```

Keyword Descriptions

symbol

```
►►─┬──────────┬─►◄
   └─symbol──┘
```
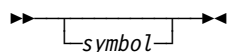
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

ADDSESS

```
                ┌─ADDSESS=0──────┐
►►──────────────┼────────────────┼──►◄
                └─ADDSESS=count──┘
```
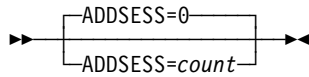
Default: 0
Minimum: 0
Maximum: 65535

Specifies how many unreserved LU-LU boundary session control blocks (BSBs) are to be made available for use by any independent logical unit in addition to those reserved by RESSCB on the LU definition statement.

When GENILU=NO is specified on the BUILD definition statement, NDF increases the value of ADDSESS by the value of RESSCB on each LU definition statement with LUDR=YES that defines an independent logical unit.

Note: You can define part of the NCP buffer pool to be used for dynamically created LU-LU boundary session control blocks when resources explicitly defined by ADDSESS on the BUILD definition statement have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* and the *NCP and EP Reference* for more information about dynamically created control blocks. See the control block management information in the *NCP and EP Reference* for details about which control blocks get created for the ADDSESS keyword.

AUXADDR

```
                ┌─AUXADDR=0──────┐
►►──────────────┼────────────────┼──►◄
                └─AUXADDR=count──┘
```
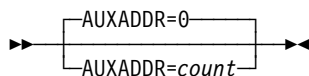
Default: 0
Minimum: 0
Maximum: 32762

Specifies how many additional addresses can be assigned to all peripheral logical units. One network address is automatically assigned for each logical unit you define. If any logical unit requires more than one address (independent logical units may have more than one primary logical unit (PLU) address), code AUXADDR to indicate the total number of additional addresses required by all logical units.

Include room for control session overhead (such as the SNA service manager) in the value of AUXADDR. These sessions can exist between type 2.1 physical unit nodes.

Note: You can define part of the NCP buffer pool to be used for dynamically created resources for additional network addresses when resources explicitly defined by the AUXADDR keyword have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP,*

SSP, and EP Resource Definition Guide and the *NCP and EP Reference* for more information about dynamically created control blocks. For more information about which control blocks get created for the AUXADDR keyword, see the *NCP and EP Reference*.

BACKUP

```
                      ┌─BACKUP=0────────┐
►►────────────────────┼─────────────────┼────►◄
                      └─BACKUP=count────┘
```
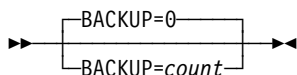
Use: Extended recovery facility (XRF)
Default: 0
Minimum: 0
Maximum: 65535

Specifies how many logical units will be available for XRF support. NCP provides XRF support when you specify a value for *count* that is greater than 0. NCP also provides XRF support using dynamically created control block resources for XRF backup if you code BACKUP=0 and code DYNPOOL on the BUILD definition statement with a nonzero value. For more information about how to define part of the NCP buffer pool to be used for dynamically created backup resources when explicitly defined resources have been exhausted, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

Notes:

1. Avoid coding excessive values for BACKUP that can waste NCP storage.

2. If you code a nonzero value for DYNPOOL on the BUILD definition statement and omit BACKUP from your generation definition, NCP does not provide XRF support.

BFRS

```
                      ┌─BFRS=(240,100)──────────────┐
►►────────────────────┼─────────────────────────────┼────►◄
                      │        ┌─240─┐     ┌─100──┐  │
                      └─BFRS=(─┼─────┼──,──┼──────┼─)┘
                               └─size┘     └─count┘
```
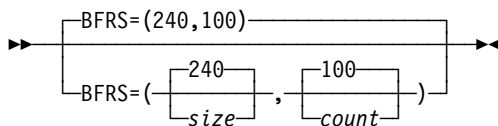
| Use: NCP or EP

Defines the size of the buffers in the NCP buffer pool and specifies the number of EP buffers to be allocated to the EP free buffer pool.

size

Default:	240
Minimum:	See text
Maximum:	240

Specifies the size, in bytes, of the buffers in the NCP buffer pool. Define *size* as a multiple of 4 bytes. The minimum buffer size is 72 bytes, unless indicated otherwise in the following notes.

Notes:

1. The generation process rounds *size* to the next higher multiple of 4 bytes if the value is not a multiple of 4.

2. The generation process adds 12 bytes to each buffer for control use. For example, if you code BFRS=240, each buffer actually uses 252 bytes. These 12 bytes are never used to contain message data.

3. The minimum size is 76 bytes when you include the online testing facility in NCP (OLT=YES on the BUILD definition statement).

4. The minimum size is 116 bytes if Internet Protocol (IP) resources are defined.

5. If you code DYNPOOL on the BUILD definition statement to include support for dynamic allocation of control blocks, the minimum buffer size is 128 bytes.

6. If this NCP supports 3745 frame-relay BAN connections, the minimum buffer size is 100 bytes.

count

Default:	100
Minimum:	100
Maximum:	500

Specifies the number of buffers to be allocated to the EP free buffer pool.

Multilink Transmission Groups: The value of BFRS can affect the segment size of multilink transmission groups (MLTGs). For more information, refer to the discussion of MLTG segment size in the *NCP, SSP, and EP Resource Definition Guide*.

BRANCH

```
          ┌─BRANCH=100─────┐
►►────────┼────────────────┼───►◄
          └─BRANCH=entries─┘
```
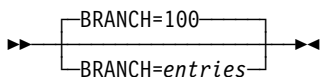
Use:	NCP or EP
Default:	100
Minimum:	100
Maximum:	8000

Specifies how many entries the branch trace table is to contain.

CAIOTRC
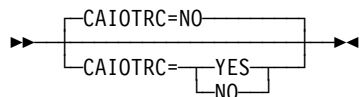
```
                  ┌─CAIOTRC=NO─────────┐
  ►►──────────────┴────────────────────┴──►◄
                  └─CAIOTRC=─┬─YES─┬────┘
                            └─NO──┘
```

| Use: NCP or EP
          Default: NO

Specifies whether NCP is to include the channel adapter I/O halfword trace.

Attention: Use this trace for diagnosis only. It can cause significant performance degradation at high utilization rates.

ESCON Links: The channel adapter I/O halfword trace does not apply to ESCON links.

CATRACE

```
                  ┌─CATRACE=(YES,50)───────────────────────────┐
  ►►──────────────┴────────────────────────────────────────────┴──►◄
           └─CATRACE=─┬─(─┬─YES─┬───,─┬─50──────┬─)─┘
                     │    └─YES─┘     └─entries─┘  │
                     └─NO──────────────────────────┘
```
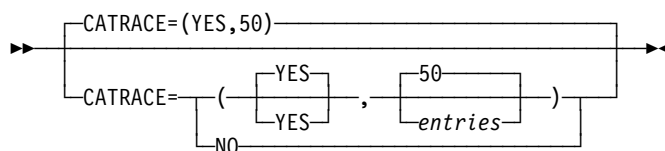
          Use: Channel adapter trace facility
          Default: YES

Specifies whether NCP is to include the channel adapter trace facility and how many entries the channel adapter trace table is to contain.

entries

> Default: 50
> Minimum: 1
> Maximum: 512

> Specifies how many 64-byte entries the channel adapter trace table is to contain.

ESCON Links: The channel adapter trace does not apply to ESCON links.

CNLSQMAX

```
                  ┌─CNLSQMAX=10000─┐
  ►►──────────────┴────────────────┴──►◄
           └─CNLSQMAX=bytes─┘
```
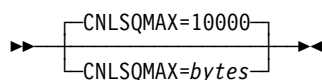
          Use: Internet Protocol
          Default: 10000
          Minimum: 1000
          Maximum: 65535

Specifies, in bytes, the maximum size for a session queue associated with an internet interface when the number of available buffers drops below the pseudo-slowdown threshold (the slowdown threshold, defined by SLODOWN on the BUILD definition statement, plus the global committed buffer count). When NCP is in this

state, frames are not placed on any internet interface outbound queue if the number of bytes queued would exceed the value of CNLSQMAX.

Frame Relay: CNLSQMAX does not apply to IP over frame relay interfaces. See IPQLIM on the PU definition statement on page 2-384 for more information about queue sizes.

Channel: CNLSQMAX does not apply to IP channel interfaces. For IP channel resources, the maximum outbound queue size is the amount of data that can be sent in three read channel programs.

CNLSQTIM

```
              ┌─CNLSQTIM=10────────┐
►►────────────┼────────────────────┼────►◄
              └─CNLSQTIM=seconds──┘
```
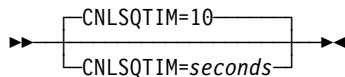
Use: Internet Protocol
Default: 10
Minimum: 1
Maximum: 255

Specifies, in seconds, how long NCP can keep a frame on an outbound session queue associated with an internet interface. When this limit is reached for a given frame, the frame is discarded.

Channel: CNLSQTIM does not apply to IP channel interfaces.

CSMHDR

```
►►──────────────────────────►◄
      └─CSMHDR=header─┘
```
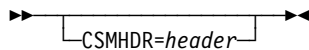
Use: BSC 3270 devices
Default: None

Defines the header of the critical situation message to be sent to any IBM 3270 terminals in the network controlled by this NCP.

Code the hexadecimal representation of the EBCDIC characters to be sent. You can specify up to 127 EBCDIC characters. If the header exceeds this length, code the CSMHDRC keyword to define the remaining characters, up to a combined total of 238 characters.

CSMHDR and CSMHDRC together can define a header up to 238 characters long, and CSMSG and CSMSGC together can define a message up to 238 characters long. However, the combined length of the header and message defined by these four keywords cannot exceed 238 characters.

CSMHDRC
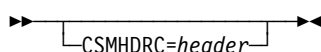
```
►►──────────────────────────────►◄
      └─CSMHDRC=header─┘
```

Use: BSC 3270 devices
Default: None

Defines up to 111 additional characters for the critical situation message header defined by the CSMHDR keyword. CSMHDRC is valid only if you code CSMHDR.

CSMSG

```
►►──┬────────────────────┬──►◄
    └─CSMSG=message─┘
```
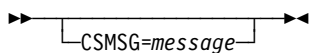
Use: Start-stop and BSC devices
Default: None

Defines the text of the critical situation message to be sent to the active stations on lines in network control mode before automatic network shutdown occurs. If you omit CSMSG, NCP does not notify stations before automatic network shutdown occurs. The message is sent to each active station defined with CRITSIT=YES on the TERMINAL definition statement.

Code the hexadecimal representation of the EBCDIC characters to be sent. You can specify up to 127 EBCDIC characters. If the header exceeds this length, code the CSMSGC keyword to define the remaining characters, up to a combined total of 238 characters.

CSMSG and CSMSGC together can define a message up to 238 characters long, and CSMHDR and CSMHDRC together can define a header up to 238 characters long. However, the combined length of the header and message defined by these four keywords cannot exceed 238 characters.

CSMSGC

```
►►──┬─────────────────────┬──►◄
    └─CSMSGC=message─┘
```

Use: Start-stop and BSC devices
Default: None

Defines up to 111 additional characters for the critical situation message defined by the CSMSG keyword. CSMSGC is valid only if you code CSMSG.

CSSTIER

```
         ┌─CSSTIER=A─┐
►►──┬──────────────────┬──►◄
    └─CSSTIER=──┬─A─┬──┘
               ├─B─┤
               └─C─┘
```

Use: 3746 Model 900 connectivity subsystem
Default: See text

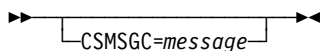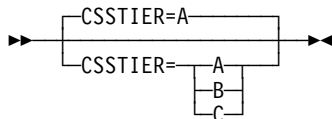Specifies the usage tier level at which you are using the optional NCP Feature for 3746 Model 900 connectivity subsystem support. The usage tier level determines how many 3746 Model 900 adapters you can use. CSSTIER=A lets you use a maximum of five 3746 Model 900 adapters; CSSTIER=B and CSSTIER=C let you use all the 3746 Model 900 adapters. For NCP V7 releases, CSSTIER=C indicates that you have the no charge feature that is available with the network node hardware feature 5022 installed on your 3746 Model 900. To determine the usage tier level you require, refer to your IBM Customer Agreement.

CSSTIER=A is the default when you specify NCP support for the 3746 Model 900 connectivity subsystem on the VERSION keyword (VERSION=V*x*R*y*F).

Note: If you code CSSTIER and you do not have the NCP 3746 Model 900 Feature, NDF will generate an error message.

NDF will issue an informational message in the generation listing when the following occurs:

- CSSTIER=A is specified and the number of adapter positions, including possible adjacent even-numbered positions in odd-even pairs, exceeds five.

  Note: 3746 Model 900 SDLC lines can be configured such that two adapter positions in an odd-even pair map to one communication line adapter (CLA), causing the total number of CLA positions used to exceed the number specified in the NCP generation definition. See "Relative Line Numbers for the 3746 Model 900 Connectivity Subsystem" on page 2-188 for more information about 3746 Model 900 CLAs.

- CSSTIER=A is specified, the TD USGTIER suffix is coded, and three or more 3746 Model 900 adapter positions are indicated in the generation definition.

  Note: When an IBM 3745-41A or 3745-61A Communication Controller operates in *twin-in-dual* mode, NDF does not know how many adapter positions will be used by the other central control unit (CCU). When you code CSSTIER=A, do not attempt to divide six or more 3746 Model 900 adapters between the two CCUs of a controller operating in twin-in-dual mode. For more information about defining controllers operating in twin-in-dual mode, see the *NCP, SSP, and EP Resource Definition Guide*.

Note: You must obtain a license for the 3746 Model 900 CSS feature. Refer to Order instructions in the NCP announcement letter for ordering information.

CUID

```
►►──┬────────────────────┬──►◄
    └─CUID=identifier─────┘
```
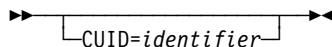
Use: Switched BSC and TWX devices
Default: None

Specifies the identification characters NCP is to send to BSC stations on switched lines operating in network control mode. NCP sends these characters to each station defined with CUIDLEN on the TERMINAL or COMP definition statement. NCP sends the characters each time it calls the station or answers a call from the station.

Code the hexadecimal representation of the EBCDIC characters to be sent. Specify a string of 20 to 40 EBCDIC characters.

If you omit CUID, NCP verifies the station identifier sequences it receives, but does not send a controller identifier sequence in return.

CWALL

```
                  ┌─CWALL=26─────┐
►►──┼──────────────┼──►◄
                  └─CWALL=buffers─┘
```
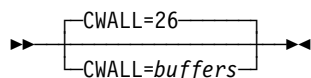
Default: 26
Minimum: 26
Maximum: 255

Specifies how many buffers NCP is to reserve to complete the flow of data through NCP. When the number of NCP buffers drops below this threshold, no more data is accepted until additional buffers become available.

DIALTO

```
                  ┌─DIALTO=60.0────┐
►►──┼──────────────┼──►◄
                  └─DIALTO=seconds─┘
```
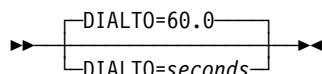
Use: Switched lines
Default: 60.0
Maximum: 1632

Specifies how long NCP is to wait for an abandon-call-and-retry (ACR) signal from an automatic calling unit. You can specify this value as an integer or with one decimal place. Use the default value of 60 seconds unless your system designer recommends a different value.

DR3270

```
►►──┬──────────────────┬──►◄
    │         ┌─YES─┐   │
    └─DR3270=─┼─────┼───┘
              └─NO──┘
```
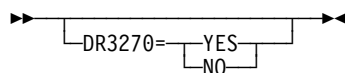
Use: Dynamic reconfiguration
Default: See text

Specifies whether NCP is to support SDLC 3270 Model 11 and Model 12 terminals even though no such terminals are defined in the generation definition. This allows you to add these terminals through dynamic reconfiguration.

The default is YES if you code a PUDRPOOL definition statement. The DR3270 keyword is ignored if you do not code PUDRPOOL.

If you code BNNSUP=3270 on the PU definition statement, NCP will support these types of 3270 terminals, so you do not need to code the DR3270 keyword.

DSABLTO

```
              ┌─DSABLTO=3.0───────┐
►►────────────┴───────────────────┴──────►◄
              └─DSABLTO=seconds──┘
```
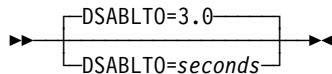
Default: 3.0
Maximum: 1632

Specifies the amount of time to wait for the data set ready (DSR) lead to drop during a disable operation. You can specify this value as an integer or with one decimal place.

DUPADDR

```
              ┌─DUPADDR=WARNING──┐
►►────────────┴───────────────────┴──────►◄
              └─DUPADDR=INFO─────┘
```
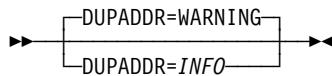
Default: WARNING

Specifies whether the ICN026I message produced for duplicate poll addresses on the PU definition statement is to be an informational message or a warning message.

DYNNSC
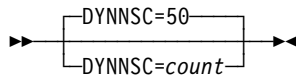
```
              ┌─DYNNSC=50────────┐
►►────────────┴───────────────────┴──────►◄
              └─DYNNSC=count─────┘
```

Use: Dynamically created control block pools
Default: 50
Minimum: 1
Maximum: 65535

Specifies the maximum number of NPM session counter control blocks (NSCs) NCP can create dynamically. The DYNNSC value limits the number of buffers used to create NSCs. DYNNSC is valid only when you code a nonzero value for the first suboperand of DYNPOOL, and you code SESSACC=YES. See DYNPOOL on the BUILD definition statement on page 2-19 for information about defining dynamically created resources. See the *NCP, SSP, and EP Resource Definition Guide* for more information about using dynamically created control blocks.

DYNNSX
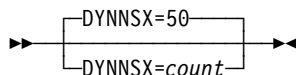
```
              ┌─DYNNSX=50────────┐
►►────────────┴───────────────────┴──────►◄
              └─DYNNSX=count─────┘
```

Use: Dynamically created control block pools
Default: 50
Minimum: 1
Maximum: 65535

Specifies the maximum number of NPM session counter extensions blocks (NSXs) NCP can create dynamically. The DYNNSX value limits the number of buffers used to create NSXs. DYNNSX is valid only when you code a nonzero value for the first suboperand of DYNPOOL, and you code SESSACC=YES. See DYNPOOL on the BUILD definition statement on page 2-19 for information about defining dynamically created resources. See the *NCP, SSP, and EP Resource Definition Guide* for more information about using dynamically created control blocks.

DYNPOOL

```
                  ┌─DYNPOOL=(0,0)──────────────────────────────────────┐
►►─────────────────┼────────────────────────────────────────────────────┼──►◄
                  └─DYNPOOL=(──┬─0──────────────┬──,──┬─0───────────────┬──)─┘
                              └─percent_of_buffers─┘   └─alert_threshold─┘
```
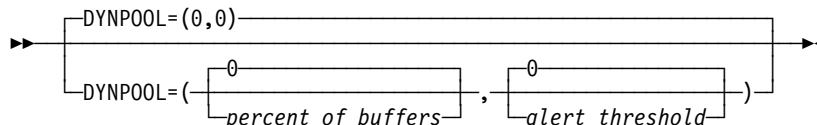
Use: Dynamically created control block pools

Defines the percent of NCP buffers available for dynamic allocation of control blocks and the associated alert threshold.

Note: Using these dynamically created resources can degrade NCP performance, so it is recommended that you define all the resources you expect to need in your generation definition and use this function only to provide control block resources for unexpected demand. For information about defining limits within dynamically created control block pools, see DYNNSC and DYNNSX on the BUILD definition statement on page 2-18. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

percent_of_buffers

Default: 0
Minimum: 0
Maximum: See text

Specifies the percentage of the buffer pool that can be used for dynamically created control blocks. When the *percent_of_buffers* is zero, none of the buffer pool may be used for dynamic allocation of control blocks. When the *percent_of_buffers* value is greater than zero, 128 bytes is the minimum valid *size* for BFRS on the BUILD definition statement. The maximum value for *percent_of_buffers* is (100 –(1.75 x SLODOWN)). If you code a *percent_of_buffers* value that exceeds the maximum, the percentage will be reduced to the calculated maximum value.

Note: The result of 1.75 times SLODOWN will be rounded up before it is subtracted from 100. For example, if SLODOWN=6, the maximum value for *percent_of_buffers* is 89.

alert_threshold

Default: 0
Minimum: 0
Maximum: Value of *percent_of_buffers* minus 1

Defines the threshold (as a percentage of the buffer pool) at which NCP is to send an alert indicating that the user-defined threshold has been reached. If you code a zero value, NCP issues no alert.

Note: When the alert threshold is reached, the NetView program does not relay the generic alert information unless it is configured to relay *impending problem* alerts.

ENABLTO

```
                  ┌─ENABLTO=2.2────┐
►►──────────────┼                 ┼──►◄
                  └─ENABLTO=seconds─┘
```
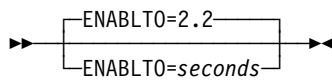
Default: 2.2
Maximum: 1632

Specifies how long NCP is to wait for a DSR signal from a modem after the line attached to the modem is enabled (for nonswitched lines), or after a dialing operation is completed (the automatic calling or answering unit has signaled a connection for switched lines). You can specify this value as an integer or with one decimal place. The enable timer is also used as the modem retrain timer. If you code a value greater than 57.7 seconds, NCP may incorrectly report a scanner error.

For a nonswitched line or a switched line where calls are made by an automatic calling unit (ACU), the default value of 2.2 seconds is usually appropriate. Generally, 2.2 seconds is also appropriate for switched lines with automatic answer. However, a longer enable time-out is required if the channel-attached NCP being defined includes any switched backup SDLC links to a link-attached controller.

ERLIMIT

```
                  ┌─ERLIMIT=8────────┐
►►──────────────┼                   ┼──►◄
                  └─ERLIMIT=─┬─8──┬──┘
                             └─16─┘
```
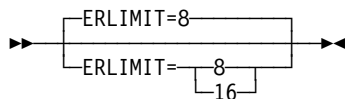
Default: 8

Specifies whether NCP is to support 8 or 16 explicit routes.

You can specify ERLIMIT for non-native networks on the NETWORK definition statement. If you code ERLIMIT on the NETWORK definition statement, that value is used for that network instead of the value specified on the BUILD definition statement.

If you code ERLIMIT on both the BUILD and NETWORK definition statements, the value coded on the BUILD definition statement applies to the native network and the value coded on the NETWORK definition statement applies to the network defined by that statement. If you do not code ERLIMIT on either definition statement, NCP uses the default value of 8 explicit routes.

GENILU

```
                        ┌─GENILU=YES─┐
►►──────┴────────────┴──────►◄
         └─GENILU=─┬─YES─┬─┘
                    └─NO──┘
```
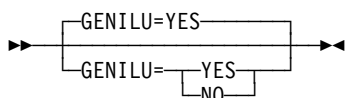
Use: Independent logical units
Default: YES

Specifies whether NDF is to generate resource control blocks and element addresses for independent logical units. For more information about defining independent logical units, see the *NCP, SSP, and EP Resource Definition Guide*.

When you code GENILU=YES or do not code GENILU, NDF generates resource control blocks and element addresses for all independent logical units defined by LU definition statements.

When you specify GENILU=NO, NDF validates all LU definition statements with LUDR=YES that define independent logical units but do not generate resource control blocks and element addresses. When GENILU=NO, NDF increases certain keyword values for each independent logical unit you define. The value for the NUMILU keyword on the LUDRPOOL definition statement is increased by one for each LU definition statement with LUDR=YES that defines an independent logical unit. The value of ADDSESS on the BUILD definition statement is increased by the value of RESSCB on each LU definition statement with LUDR=YES that defines an independent logical unit. NDF does not increment the values for NUMILU and ADDSESS beyond the maximum allowed for each keyword.

When GENILU=NO, NCP collects data for independent logical units for which LUDR=YES is specified only if NPA=(YES,DR) or NPA=(YES,DRTP) is also specified on the BUILD definition statement.

GENLEVEL

```
►►──┬──────────────────────┬──►◄
     └─GENLEVEL=*string*─┘
```
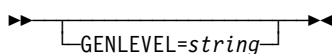
Default: See text

Attention: Do not code the GENLEVEL keyword until you read the discussion of GENLEVEL in the *NCP, SSP, and EP Generation and Loading Guide*. This information can help you avoid problems with your generation definition.

GENLEVEL enables VTAM to load and activate NCP even if NCP's date and time of generation do not match the DTG control block and the type 1 RRT entry.

Code a string up to 17 characters long. The string may contain blanks only if surrounded by quotes. The default is the date and time of NCP generation in the format GENLEVEL='mm/dd/yy hh:mm:ss'.

GWAEXIT

```
►►──┬──────────────────┬──►◄
     └─GWAEXIT=name─────┘
```
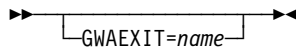
Use: Gateway NCP and user-written code
Default: None

Specifies the entry point in the module that contains the user-written accounting exit routine for cross-network sessions. *name* is the name of the entry point in the module included in the NCP and PEP load module by means of the standard attachment facility or the INCHI and ORDHI keywords on the GENEND definition statement.

GWPACING
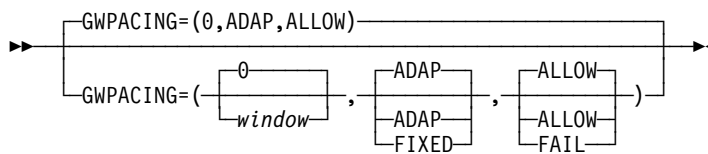
```
         ┌─GWPACING=(0,ADAP,ALLOW)──────────────────┐
►►──┬────┴──────────────────────────────────────────┴──►◄
     │                ┌─0──┐     ┌─ADAP─┐    ┌─ALLOW─┐      │
     └─GWPACING=(─┴─window─┴─,─┴─ADAP─┴─,─┴─ALLOW─┴─)─┘
                                  └─FIXED┘    └─FAIL─┘
```

Use: Gateway NCP
Example: GWPACING=(10,,FAIL)

Specifies the actions to take during a cross-network session if the BIND request or response indicates that session pacing will not control the flow of data for this session.

window

Default: 0
Minimum: 0
Maximum: 63

Specifies the pacing window size to be used to override the session pacing parameters in a negotiable BIND for SNA LU-LU cross-network sessions. The session pacing parameters on a negotiable BIND are overridden if they indicate that no session pacing is to be used on this session. If you code *window* or allow it to default to zero, the pacing parameters on a negotiable BIND are not overridden.

ADAP│**FIXED**

Default: ADAP

Determines the action to be taken on BIND for SNA LU-LU cross-network sessions that specify that adaptive session pacing may be used on the session. ADAP allows the BIND to pass through unchanged. FIXED turns off the adaptive session pacing support bit in the BIND, which prevents adaptive session pacing from being used on this session.

ALLOW│**FAIL**

Default: ALLOW

Determines the action to be taken on nonnegotiable BINDs and negotiable BIND responses for SNA LU-LU cross-network session activation that specify that no session pacing will be used. ALLOW permits the session activation attempt to proceed. FAIL does not allow the session activation to proceed.

GWSESAC

```
           ┌─GWSESAC=NO────────────────────────────────────┐
►►─────────┤                                               ├─►◄
           └─GWSESAC=──┬─(YES,─┤ Accounting parameters ├─)─┘
                       └─NO────────────────────────────┘
```

Accounting parameters:

```
     ┌─NODEFER─┐
├─────┼─DEFER───┼──,──┬───────┬──,byteth,piuth,──┬──────┬──,──┬──<─,──┬─────┬──┬──,──►
     └─NODEFER─┘      └─nscnum┘                  └─nscx─┘     └────┬─piudist─┘
                                                                  

►──┬──────────────┬─────►
   └─INITialpcid──┘
```
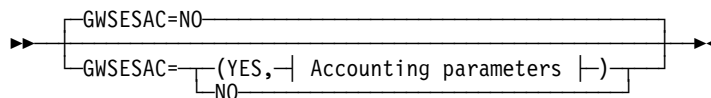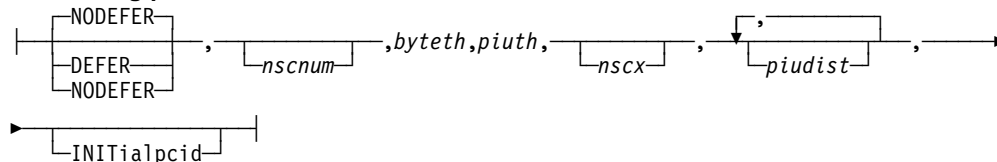
Default: NO
Example: GWSESAC=(YES,,,65535,65535)

Specifies whether you want to include the gateway session accounting function. GWSESAC=YES is valid only if you have NetView Performance Monitor (NPM) R3 or a later release. GWSESAC=YES is valid only if SESSACC=YES and HSBPOOL are specified.

NODEFER│DEFER

  Default: NODEFER

  Indicates whether you want gateway session accounting to begin immediately or be deferred until NCP receives a change-accounting PIU.

nscnum

  Default: See text
  Minimum: 0
  Maximum: 65535

  Specifies how many NPM session accounting blocks (NSCs) are to be gener-ated in addition to those generated for session accounting by the SESSACC keyword. The default is half the sum of the following:

  - The value of HSBPOOL on the BUILD definition statement
  - The sum of the values of NUMSESS on all GWNAU definition statements.

  If the total number of NSCs defined by the SESSACC and GWSESAC keywords exceeds 65 535, NDF generates 65 535 NSCs.

  Note: You can define part of the NCP buffer pool to be used for dynamically created NSCs when NSCs explicitly defined by GWSESAC and SESSACC on the BUILD definition statement have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See DYNNSC on the BUILD definition statement on page 2-18 for information about defining limits within dynamically created control block pools and see the *NCP, SSP, and EP Resource*

Definition Guide for more information about dynamically created control blocks.

byteth

Use:	Required if you code GWSESAC
Default:	None
Minimum:	1
Maximum:	4294967295 ($2^{32}-1$)

Specifies how many bytes NCP is to count before sending gateway session accounting data to NPM. *byteth* must be greater than or equal to *piuth*.

Note: The value 2147483648 (2^{31}) is invalid.

piuth

Use:	Required if you code GWSESAC
Default:	None
Minimum:	1
Maximum:	65535

Specifies how many PIUs NCP is to count before sending gateway session accounting data to NPM. *piuth* must be less than or equal to *byteth*.

Attention: Do not choose values for *byteth* and *piuth* that are too small. Small values have an adverse effect on NCP performance because too much time is spent sending data to NPM.

nscx

Default:	See text
Minimum:	0
Maximum:	65535

Specifies how many NSC extensions are to be generated in addition to those generated for session accounting by the SESSACC keyword. Zero may be specified, but is not recommended. The default is 3 or *nscnum* × 1.3, whichever is greater.

If the total number of NSC extensions defined by the SESSACC and GWSESAC keywords exceeds 65 535, NDF generates 65 535 NSC extensions.

When session accounting records are delayed, NSC extensions are needed to save information that otherwise would be lost. Sessions that use a procedure-correlation identifier (PCID) normally require NSC extensions. When a session is being defined to a gateway NCP, a PCID may be passed to the gateway NCP. When this occurs, an NSC extension is required regardless of whether gateway session accounting is active. An NSC extension is released upon establishment of a session for which no accounting is performed. Therefore, the NSC extensions for each session may be used either for the PCID or for information to be saved when session accounting records are delayed.

Note: You can define part of the NCP buffer pool to be used for dynamically created NSC extensions when NSC extensions explicitly defined by GWSESAC and SESSACC on the BUILD definition statement have

been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See DYNNSX on the BUILD definition statement on page 2-18 for information about defining limits within dynamically created control block pools and see the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

piudist

Default: None
Minimum: 0
Maximum: 65535

Specifies up to six PIU distribution lengths by which PIUs flowing through a particular session are to be counted. Code *piudist* values in ascending order.

Code at least one *piudist* value if you want to be able to specify PIU distribution ranges at a later time. If you do not define any PIU distribution ranges, NCP does not include storage for the distribution counters.

If you code a *piudist* value of 0, all subsequent *piudist* values are ignored.

INITialpcid
Specifies that the PCID used to allocate NCP SNI resources when setting up a session is not to be overridden by the PCID used to establish the session when the session is bound.

HPR
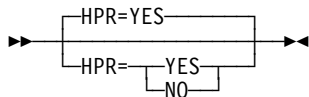
Default: YES

Specifies whether you want NCP to include the high performance routing (HPR) function. Valid values are YES and NO.

- HPR=YES means include the HPR function
- HPR=NO means do not include the HPR function

Note: If you are not going to use HPR, coding HPR=NO causes the unique HPR code to not be included in the NCP load module. This leaves more storage available for NCP buffers and so forth.

A warning message is issued for the following HPR keywords if you code HPR=NO on the BUILD definition statement.

- HPRMPS on the BUILD definition statement
- HPRMLC on the BUILD definition statement
- HPRATT on the BUILD definition statement
- HPRQLIM on the BUILD, GROUP, LINE, or PU definition statements
- LLERP on the GROUP, LINE, or PU definition statements

HPRATT

```
                    ┌─HPRATT=12000─┐
►►──────────────────┼──────────────┼──────►◄
                    └─HPRATT=nnn───┘
```
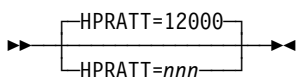
Use: Composite automatic network routing (ANR) node's subarea network
              flow control
Default: 12000
Minimum: 0
Maximum: 2147483647 (2^{31}-1)

Specifies the average time in microseconds that is required to transmit 1200 bits
across a typical path in the composite ANR node's subarea network. (The Rapid
Transport Protocol (RTP) nodes use the accumulated transmission time of all the
nodes and links along an RTP connection to determine the rate at which data will
be sent on the RTP connection.)

NCP estimates this time for the subarea network if the time is greater than or equal
to 200 000 microseconds. If the time is less than 200 000 microseconds, code a
value on the HPRATT keyword. The value you provide is used when transmission
begins. After transmission has begun, the actual value, which the RTP endpoints
track, is used.

To estimate the value to use, determine the time it takes to send 1200 bits across a
typical path in the subarea network. Table 2-4 shows the times you should use for
the different line types and speeds for times less than 200 000 microseconds. If
the typical path traverses more than one subarea node, use the sum of all the links
along the path.

Table 2-4 (Page 1 of 2). Accumulated Transmission Time by Line Type and Speed		
Line Type	**Line Speed (bits per second)**	**Accumulated Transmission Time (microseconds)**
Token Ring	—	776 (TIC1) 362 (TIC2) 90 (TIC3)
ESCON Channel	—	90
Type 6 Channel	—	390
Type 7 Channel	—	203

Table 2-4 (Page 2 of 2). Accumulated Transmission Time by Line Type and Speed		
Line Type	**Line Speed (bits per second)**	**Accumulated Transmission Time (microseconds)**
SDLC, Frame Relay, or X.25 ODLC	9600	125000
	19200	62500
	38400	31250
	55855	21485
	64000	18750
	245760	4883
	256000	4688
	512000	2344
	1024000	1172
	1474560	814
	1536000	782
	1843200	652
	2048000	586
ISDN	64000	18750

For example, a link that functions at 9600 bits per second would use a value of 125000 for *nnn*.

To calculate transmit time across an MLTG, use the transmit time of the slowest link in the MLTG, **NOT** the sum of the transmit times of all the links.

The default of 12000 microseconds was derived by dividing 1200 bits by a transmission rate of 100 000 bits per second.

HPRMLC

```
        ┌─HPRMLC=9───┐
►►──────┤            ├──────►◄
        └─HPRMLC=nnn─┘
```
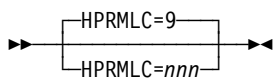
Use: Composite ANR node's subarea network flow control
Default: 9
Minimum: 0
Maximum: 65535

Specifies the capacity in kilobits per second of the slowest subarea transmission group in the composite ANR node's subarea network that can carry APPN HPR data. The default is 9, which is for 9.6 kilobits per second.

Use Table 2-5 on page 2-28 to help you find the slowest subarea transmission group in the subarea network.

To calculate the capacity of an MLTG, sum up the capacities of the individual links in the TG.

Table 2-5. Minimum Link Capacity by Line Type and Speed		
Line Type	**Line Speed (bits per second)**	**Minimum Link Capacity (kilobits per second)**
Token Ring	—	1510 (TIC1) 3238 (TIC2) 13062 (TIC3)
ESCON Channel	—	13062
Type 6 Channel	—	3008
Type 7 Channel	—	5760
SDLC, Frame Relay, or X.25 ODLC	75	1
	100	1
	110	1
	134	1
	200	1
	300	1
	600	1
	1200	1
	2400	2
	4800	4
	9600	9
	19200	18
	38400	37
	55855	54
	64000	62
	245760	240
	256000	250
	512000	500
	1024000	1000
	1474560	1440
	1536000	1500
	1843200	1800
	2048000	2000
ISDN	64000	62

HPRMPS

```
        ┌─HPRMPS=0───┐
►►──────┤            ├──────►◄
        └─HPRMPS=nnn─┘
```
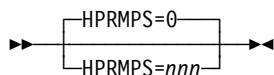
Use: Composite ANR node's subarea network flow control
Default: 0
Minimum: 768
Maximum: 65535

Specifies the largest packet that can be sent across the composite ANR node's subarea network without being segmented on any of the subarea links along the path.

If all the NCPs in the subarea network are at V7R3 or later, code 0 for the value of HPRMPS, or let it default to 0. The correct HPRMPS value will be calculated.

If any of the NCPs in the subarea network are at V7R2 or earlier, code HPRMPS. It can only be coded in a V7R3 or later NCP. If the HPRMPS value is too large, APPN HPR packets may be segmented as they flow across the subarea network, which increases processing in the composite ANR node.

Note: An HPRMPS value for the V7R3 or later NCPs in the subarea network is calculated. But, this will not include NCPs that are at V7R2 or earlier. The HPRMPS value you code on the HPRMPS keyword is compared to the value calculated for the V7R3 or later NCPs, and the smaller of the two values is used.

To determine what value to use if any of the NCPs in the subarea network are at V7R2 or earlier, find the smallest PIU segment size supported in the subarea network of the composite ANR node. Table 2-6 shows what segment sizes are used for each line protocol:

Table 2-6. Segment Sizes Used by NCP

Line Protocol	Keywords that Affect Segment Size	How to Calculate Segment Size
SDLC	TRANSFR, BFRS	(TRANSFR * BFRS)−18 bytes [1]
Token Ring	MAXTSL	MAXTSL value [2]
NTRI	RCVBUFC	RCVBUFC value−36 bytes [3]
Frame relay or ISDN	MAXFRAME	MAXFRAME value−16 bytes [4] MAXFRAME value−32 bytes or LF-4 or MAXTSL - 36 (BAN) [4]

[1] The PIU segment size for an SDLC link depends on the TRANSFR value and BFRS value of the adjacent NCP, which are transferred when the XID exchange takes place. Multiply the TRANSFR value of the adjacent NCP by the BFRS value and subtract 18 bytes from the result.

[2] The PIU segment size for a 3746 Model 900 Token Ring or NTRI depends on the MAXTSL value.

[3] When you define a NTRI link, you can code RCVBUFC to limit the maximum segment size to be received. NCP uses the smaller of the MAXTSL value of this node or the RCVBUFC−36 value of the adjacent node as the segment size of the NTRI link.

[4] The PIU segment size for a frame-relay or ISDN BAN link is the smaller of (1) this link's MAXFRAME−32 or LF−4, whichever is smaller, or (2) the adjacent node's LF−4 or MAXTSL−36, if the adjacent connection is to an NCP, whichever is smaller. LF is the largest frame indicator field in the routing information field in a frame-relay BAN frame. For any other frame-relay or ISDN link, the PIU segment size is the smaller of (1) this link's MAXFRAME−16 or (2) the adjacent node's MAXFRAME−16.
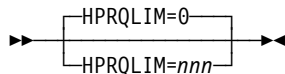
HPRQLIM

```
                  ┌─HPRQLIM=0───┐
►►────────────────┼─────────────┼──────►◄
                  └─HPRQLIM=nnn─┘
```

Use: Composite ANR node's subarea network flow control
Default: 0
Minimum: 0
Maximum: 16777215

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 its 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.

Note: A small HPRQLIM defined for a boundary channel-link might eliminate the effectiveness of the channel attention delay (DELAY) because NCP will defeat the blocking delay if the HPR queue limit is exceeded.

HPRSNAP

```
              ┌─HPRSNAP=NONE────────────────────┐
►►────────────┼─────────────────────────────────┼──────►◄
              │              ┌──,──────┐          │
              └─HPRSNAP=──┬─(─▼─┬─SETUP─┬─)──────┘
                          │      └─NLP──┘         
                          └─NONE─────────────────
```
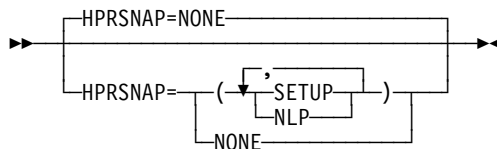
Use: APPN HPR
Default: NONE

Specifies whether the snap trace function is to be enabled for APPN HPR resources. Code SETUP to specify that NCP is to enable the snap trace function for route setup requests. Code NLP to specify that NCP is to enable the snap trace function for network layer packets (NLPs) received or routed by NCP. You can code SETUP or NLP, or both, in any order.

Note: Using the snap trace for HPR resources degrades system performance; use it only when directed to do so by IBM support personnel.

HSBPOOL

```
►►────────────────────────►◄
   └─HSBPOOL=hscbs─┘
```
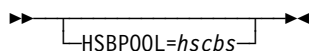
Use: Gateway NCP
Default: None
Minimum: 0
Maximum: 65535

Specifies how many half-session control blocks (HSCBs) the gateway NCP is to have for the system services control points (SSCPs) and logical units in any network that can participate in cross-network sessions. These SSCPs and LUs

include those in the native network and in all other networks that the gateway NCP joins.

NCP includes the SNA SNI function to provide gateway capability whenever you code a value on HSBPOOL. You may need to allocate more storage for NDF if you specify a large value for HSBPOOL.

Note: You can define part of the NCP buffer pool to be used for dynamically created HSCBs when explicitly defined HSCBs have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

IPPOOL

Use: Internet Protocol
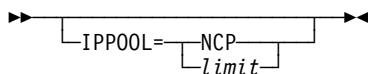Default: None
Minimum: 0
Maximum: 100

Limits the number of NCP buffers that can be used to receive IP datagrams. When the amount of internet data exceeds this limit, the excess IP datagrams are discarded. If you do not code IPPOOL, NCP can use all buffers for IP datagrams up to the slowdown threshold specified by SLODOWN on this BUILD definition statement.

Limiting the number of buffers for IP datagrams can help prevent internet traffic from interfering with SNA traffic. If you do not limit the number of buffers for IP datagrams, internet traffic can use enough NCP buffers to cause NCP to enter slowdown mode even when SNA traffic is light.

limit
    Specifies the internet buffer limit as a percentage of the number of buffers over the slowdown threshold. For more information, see SLODOWN on the BUILD definition statement. If you specify a value that equals less than 6144 bytes of buffer storage, NCP uses 6144 bytes as the limit.

NCP
    Specifies that NCP is to determine the internet buffer limit automatically.

    For NCP V7R5 and earlier releases, NCP calculates the internet buffer limit as follows:

    $$(6144 + qsize + (3/4 \times qsize \times (qcount - 1))) \div bufsize$$

    where:

    qsize is the maximum size of the NCST session queues as defined by CNLSQMAX on the BUILD definition statement.

   *qcount* is the number of NCST logical units in this NCP.

    bufsize is the NCP buffer size as defined by BFRS on the BUILD definition statement.

NCP uses up to a maximum of 25% of the buffers over the slowdown threshold. If this figure is not appropriate for your NCP, specify a different limit.

Starting in V7R6, NCP calculates the internet buffer limit as 25% of the buffers over the slowdown threshold. If this figure is not appropriate for your NCP, specify a different limit.

IPRATE

```
►►──┬───────────────────────────┬──►◄
    └─IPRATE=(rate─┬──────────┬─)─┘
                   └─,burst─┘
```
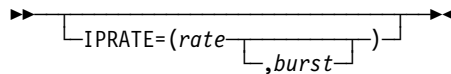
Use: Internet Protocol

Limits the rate at which NCP accepts internet frames. When the number of internet frames exceeds this rate, excess internet frames are discarded while only ARP frames continue to be accepted. If you do not code IPRATE, NCP accepts internet frames at any rate.

Defining a maximum internet frame rate can help prevent internet traffic from interfering with SNA traffic. Because internet traffic has priority over SNA traffic, SNA traffic can be impeded when the internet traffic is heavy. Limiting the rate at which NCP receives internet frames helps make NCP processing available for SNA traffic regardless of the level of internet traffic.

rate

> Default: None
> Minimum: See text
> Maximum: 65535

Specifies the maximum number of internet frames per 10th of a second that NCP is to receive at a steady rate. Table 2-7 on page 2-33 lists the approximate maximum rates at which the different IBM 3745 Communication Controller models can accept and forward internet frames from an Ethernet-type LAN. Your actual optimal maximum rate depends upon your configuration. It is suggested that you start with the value shown here for your controller and adjust it later if necessary to improve the performance of your configuration.

Note: The following maximum throughputs were developed for Ethernet. NCP supports IP for other data link control types for which the maximum throughput might be different. To establish the maximum throughput for a specific environment, the performance model contained in the 3745 configurator on HONE/IBMLINK (CF3745) can be exercised.

Table 2-7. Approximate Maximum Internet Frame Rates for an Ethernet-type LAN

| Controller Model | Maximum Frame Rate (Frames per 0.1 Second) | |
	68-Byte Packets	1500-Byte Packets
3745-130 3745-150 3745-160 3745-170 3745-17A	90	70
3745-210 3745-21A 3745-410 3745-41A	110	90
3745-310 3745-31A 3745-610 3745-61A	180	140

> **Channel:** Ensure that the value you specify for *rate* is sufficient based on the size of the write channel program.

burst

Default: Value of *rate*
Minimum: Value of *rate*
Maximum: 65535

Specifies the maximum number of internet frames that NCP is to receive for a burst of 0.1 second. When *rate* and *burst* have the same value, no bursts beyond *rate* are accepted.

Whenever the actual internet frame rate is below *rate*, the unused frame allowance is accumulated up to a maximum of *burst*. This accumulated allowance is used to accommodate 0.1-second bursts above *rate*. For example, if you specify 100 for *rate* and 150 for *burst* and NCP receives 90 internet frames per 0.1 second for 0.2 second, NCP accumulates an allowance of 20 frames (10 frames per 0.1 second for 0.2 second). This enables NCP to receive up to 120 frames during the subsequent 0.1-second interval (the 100 frames defined by *rate* plus the allowance of 20 frames).

IPSNAP
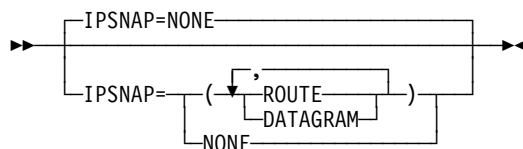
```
                     ┌─IPSNAP=NONE────────────────┐
►►──────┬─────────────────────────────────┬──────►◄
        │              ┌──,──┐             │
        └─IPSNAP=──┬─(─▼─┬─ROUTE───┬─)─┬───┘
                   │     └─DATAGRAM─┘   │
                   └─NONE──────────────┘
```

Use: Internet Protocol
Default: NONE

Specifies whether the snap trace function is to be enabled for internet resources. Code ROUTE to specify that NCP is to enable the snap trace function for internet route activity. Code DATAGRAM to specify that NCP is to enable the snap trace

function for IP datagrams received or routed by NCP. You can code ROUTE or
DATAGRAM, or both, in any order.

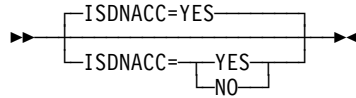**Note:** Using the snap trace for internet resources degrades system performance;
use it only when directed to do so by IBM support personnel.

ISDNACC

```
        ┌─ISDNACC=YES──────┐
►►──────┴──────────────────┴──►◄
        └─ISDNACC=─┬─YES─┐
                   └─NO──┘
```

Use: ISDN
Default: See text

Specifies whether the CSS is to include ISDN accounting information. This ISDN
data provides information about ISDN B-channel connections, for example, the
duration of a call. ISDNACC=YES is the default when SESSACC=YES is coded.
Code ISDNACC=NO to disable ISDN accounting. ISDNACC=YES is valid only if:

- NPA=YES on the BUILD statement
- SESSACC=YES on the BUILD statement
- The generation defines an ISDN physical line

ITEXTTO

```
        ┌─ITEXTTO=NONE─────┐
►►──────┴──────────────────┴──►◄
        └─ITEXTTO=─┬─seconds─┐
                   └─NONE────┘
```
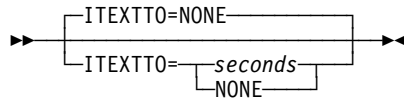
Use: Start-stop and BSC devices in network control mode
Default: NONE
Maximum: 1632

Specifies, in seconds, the text time out interval NCP is to use for any terminal or
component defined with INHIBIT=TEXTTO on the TERMINAL or COMP definition
statement. ITEXTTO=NONE specifies that no time outs are to occur.

LENAME

```
        ┌─LENAME=INLINKED─┐
►►──────┴─────────────────┴──►◄
        └─LENAME=name─────┘
```
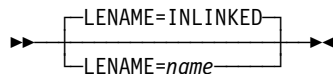
Use: VSE (NCP or EP)
Default: INLINKED

Specifies the name of the member containing the link-edit control statements.
Specify LENAME only if TYPSYS=VSE or TYPSYS=DOS. You can define any
valid symbol up to 8 characters long.

The name on the link-edit step INCLUDE statement must match the value you
specify for LENAME. Code LENAME only when NDF concurrently catalogs link-
edit cards to the same sublibrary.

LINETRC

```
                    ┌─LINETRC=(YES,4,4)─────────────────────────────────┐
►►──┼───────────────────────────────────────────────────────┼──►◄
     └─LINETRC=─┬─(─┬─YES─┬──,─┬──4────┬──,─┬──4─────┬──)─┘
               │    └─YES─┘     └─sits─┘     └─blocks─┘
               └─NO───────────────────────────────────┘
```

Use: BSC and start-stop devices in emulation mode
Default: YES

| Specifies whether the line trace and scanner interface trace (SIT) is included for
| links operating in emulation mode.

sits

  Default: 4
  Minimum: 1
  Maximum: 16

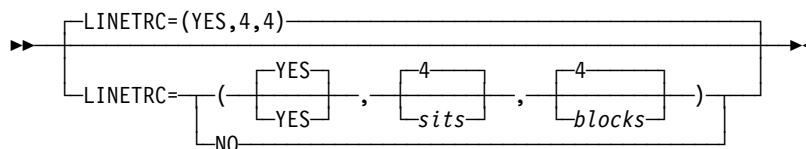 Specifies the maximum number of SITs that can be active at one time.

blocks

  Default: 4 (252 entries)
  Minimum: 2 (126 entries)
  Maximum: 400 (25 200 entries)

  Specifies how many 512-byte blocks are to be included in the line trace table.
  Each block contains 63 entries.

LOCALTO
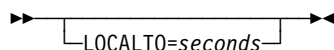
```
►►──┬──────────────────────┬──►◄
     └─LOCALTO=seconds─┘
```

Use: Token-Ring logical peripheral resources and NTRI frame handler
      logical lines
Default: See text
Minimum: 0.6
Maximum: 20.0

Specifies the Token-Ring reply timer (T1) value for stations connected to a local
Token Ring. The value must be less than or equal to the value of REMOTTO.
You can also specify LOCALTO using the T1TIMER keyword. See T1TIMER on
the BUILD definition statement on page 2-56 for additional information.

The default is either 1.5 or the value of REMOTTO, whichever is less.

LTRACE

```
                    ┌─LTRACE=2─────┐
►►──┼──────────────────┼──►◄
     └─LTRACE=count─┘
```
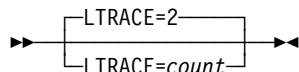
Default: 2

Minimum: 2
Maximum: 8

Specifies the maximum number of line traces and scanner interface traces (SITs) NCP is to run concurrently.

The line trace facility is a service aid that NCP always includes. NCP supports the line trace for communication lines controlled by IBM special products or user-written code. Line traces are requested from the host processor. See the *NCP and SSP Customization Guide* for further information.

MAXSESS

```
          ┌─MAXSESS=65535───┐
►►────────┼─────────────────┼────►◄
          └─MAXSESS=sessions─┘
```

Default: 65535
Minimum: 1
Maximum: 65535

Specifies the maximum number of LU-LU sessions that a boundary logical unit can have in any single NCP.

Notes:

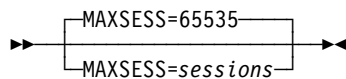1. If an independent logical unit is able to establish sessions over more than one link station, the independent logical unit can have the number of sessions specified by MAXSESS=*sessions* on each link station.

2. Be sure to include room for control session overhead (for example, SNA service manager) in the value of MAXSESS. These sessions can exist between type 2.1 physical unit nodes.

MAXSSCP

```
►►──┬──────────────┬──►◄
    └─MAXSSCP=sscps─┘
```
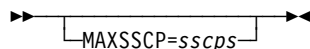
Default: See text
Minimum: See text
Maximum: 8

Specifies the maximum number of SSCPs with which NCP can be in session concurrently. These sessions can be conducted over channel adapters, SDLC communication links, or both.

NCP can be in session with as many as eight SSCPs concurrently or as few as one SSCP. However, the minimum value of MAXSSCP is the total number of active channel adapters (NCPCA=ACTIVE on the LINE definition statement). If you do not code NCPCA, the minimum value of MAXSSCP is 1.

The default is 2 if you code TYPGEN=NCP-R on the BUILD definition statement; otherwise, the default is the total number of active channel adapters. Code MAXSSCP when you want SSCP sessions to take place over SDLC links.

If you code MONLINK=CONTINUOUS, SMMF continuously monitors the link. SMMF acts as one of the SSCPs in session with NCP.

MAXSUBA

```
▶▶──┬──────────────────┬──▶◀
    └─MAXSUBA=address──┘
```
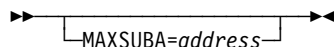
Use: Required unless all nodes in the network support extended network
            addressing
Default: None
Minimum: 3
Maximum: 255

Specifies the upper limit of the range of subarea addresses used within the native
network by NCPs that do not support extended network addressing.

The valid values for MAXSUBA are 3, 7, 15, 31, 63, 127, and 255. If you specify
any other value less than 255, NCP uses the next higher value.

The value of MAXSUBA for this NCP must match the values of MAXSUBA for all
other NCPs, EPs, and access methods in the same network.

The value of MAXSUBA does *not* affect the maximum number of resources that
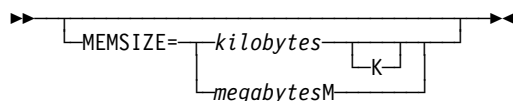can be associated with a subarea address in the network. The maximum number
of resources that can be associated with a subarea in the network is 65 535.

| **EP Standalone:** MAXSUBA is not valid.

MEMSIZE

```
▶▶──┬────────────────────────────────────┬──▶◀
    └─MEMSIZE=─┬─kilobytes─┬─────────┬──┤
              │           └─K─┘     │
              └─megabytesM──────────┘
```

| Use: NCP or EP
Default: The amount of storage installed in the controller
Minimum: 512K or 1M
Maximum: See text
Examples: MEMSIZE=2048 or MEMSIZE=2048K
            MEMSIZE=2M

Specifies the maximum amount of controller storage in KB (KB equals 1024 bytes)
| or MB (MB equals 1 048 576 bytes) that NCP or EP is to use, including storage for
| the load modules and buffers created after NCP or EP is loaded into the communi-
cation controller. NCP uses the value of MEMSIZE or the amount of storage
installed in the controller (whichever is less) as the amount of available storage in
the controller.

Note: The recommended method to use the full storage capacity of the controller
is to omit the MEMSIZE keyword.

For the IBM 3745-31A or 3745-61A communication controller, the maximum is
16384K or 16M. For any other model, the maximum is 8192K or 8M.

If specified in KB, the value of MEMSIZE must be a multiple of 128. If specified in
MB, the value of MEMSIZE must be an integer. The letter *M* is required when you
specify the value in MB.

Executable code must reside in the low 4MB of storage. The load module includes control blocks that can reside above 4MB. Other data areas and buffers that are dynamically built at load module initialization time occupy the remaining storage to the controller maximum.

MLTGORDR

```
                   ┌─MLTGORDR=FIFO─────────┐
►►─────────────────┤                       ├──────►◄
                   └─MLTGORDR=─┬─FIFO────┬──┘
                               └─MLTGPRI─┘
```
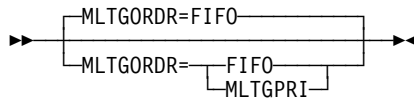
Default: FIFO

Specifies whether individual lines in MLTGs are to be selected for use in the order in which they were activated or in order of user-defined line priority.

FIFO
>   If MLTGORDR=FIFO is specified or if the default for MLTGORDR is taken, the stations are added to the list of operative stations by the order of station activation. The first station activated becomes the first station in the list and the last station activated becomes the last station in the list.
>
>   When MLTGORDR=FIFO is in effect, the fixed-order search scheme is used. The fixed-order search scheme always begins with the first station in the list of operative stations. In other words, the fixed-order search always begins with the first station activated.
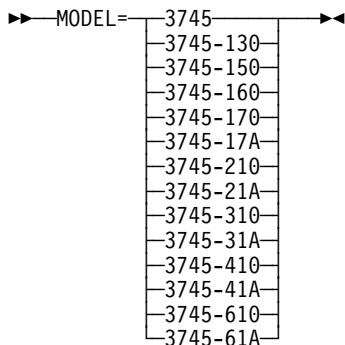
MLTGPRI
>   If MLTGORDR=MLTGPRI is specified, the stations are added to the list of operative stations based on the MLTGPRI value coded for a particular station. The station with the highest value specified for MLTGPRI becomes the first station in the list and the station with the lowest value specified for MLTGPRI becomes the last station in the list. Stations with equal MLTGPRI values are initially ordered within the list of operative stations by the order of station activation.
>
>   When MLTGORDR=MLTGPRI is in effect, the fixed/round-robin search scheme is used. This means that:
>
>   • The search for an available station always begins with one of the stations that has the highest value specified for MLTGPRI.
>
>   • If the MLTG includes more than one station with the highest value specified for MLTGPRI and if one of these stations is selected, the next search for an available station starts with one of these highest priority stations but not the one that was just selected.
>
>   • If a station is selected that does not have the highest MLTGPRI value, the next search for an available station begins with the same highest priority station that was used as the starting point for the current search.
>
>   • If a station is selected that does not have the highest MLTGPRI value and if the MLTG includes more stations with the same MLTGPRI value as that of the just selected station, the operative list of stations is reordered so that the just selected station becomes the last station, of its MLTGPRI value, to be checked for availability during the next search.

3746 Model 900:   Specifying MLTGORDR=MLTGPRI causes a more even distrib-
ution of data across the links within the MLTG. Evenly distributed data
normally produces better response time.

MODEL

```
►►──MODEL=──┬─3745─────┬──►◄
            ├─3745-130─┤
            ├─3745-150─┤
            ├─3745-160─┤
            ├─3745-170─┤
            ├─3745-17A─┤
            ├─3745-210─┤
            ├─3745-21A─┤
            ├─3745-310─┤
            ├─3745-31A─┤
            ├─3745-410─┤
            ├─3745-41A─┤
            ├─3745-610─┤
            └─3745-61A─┘
```

| Use: | Required (NCP or EP) |
| --- | --- |
| Default: | None |

Specifies the IBM communication controller model for which this NCP or EP is to
be generated.  If you code a specific model number, NDF validates your configura-
tion for that model.  NCP or EP might still be able to run on a different model if the
configuration is compatible with the specified model.

If you specify MODEL=3745, NDF does not perform validation for a specific model.
NCP or EP can run on any model.

**Note:**   When the generation is being done for an NCP which utilizes a D/T3746
model 900, MODEL=3745 is not valid.  You must specify the complete model
number.  This limitation is required because of hardware configuration variances
between single and dual CCU model 3745s.

## MTARTO

```
       ┌─MTARTO=1.0──────┐
►►─────┼─────────────────┼──►◄
       └─MTARTO=seconds──┘
```
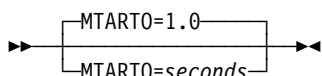
Use:	Multiple-terminal-access lines
Default:	1.0
Minimum:	0.1
Maximum:	None

Specifies, in seconds, the reply time out interval to be used when a start-stop ter-
minal on a switched multiple-terminal-access line calls the controller. You can
specify this value as an integer or with one decimal place.

MTARTRY

```
       ┌─MTARTRY=0───────┐
►►─────┼─────────────────┼──►◄
       └─MTARTRY=retries─┘
```
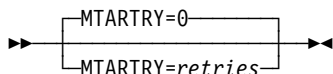
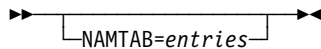| Use:	Multiple-terminal-access lines
Default:	0

Minimum: 0
Maximum: 255

Specifies the maximum number of times NCP is to retry the start-stop multiple-terminal-access sign-on procedure after identifying the type of MTA terminal that called this controller.

NAMTAB

Default: See text
Minimum: See text
Maximum: 65535

Specifies how many entries the network names table (NNT) is to contain. The names stored in this table are the unique names received in various session activation PIUs, which are SSCP names, CP names, and network names.

The minimum value for NAMTAB is the number of unique network identifiers coded using the NETID keyword in your generation. The default is 30 or the number of unique network names defined in the generation definition, whichever is greater. Code a value for NAMTAB that includes all unique network identifiers and the value coded for COPIES on the NETWORK definition statement. Refer to *NCP, SSP, and EP Resource Definition Guide* for more information on how to calculate an appropriate value.

APPN: When you define an Advanced Peer-to-Peer Networking (APPN) network, code a value for NAMTAB that is large enough to allow multiple and parallel sessions among independent logical units. If the value of NAMTAB is too small, the activation of the LU-LU session or of a physical unit with PUTYPE=2 and XID=YES may fail.

Note: You can define part of the NCP buffer pool to be used for dynamically created NNT resources when explicitly defined NNT resources have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

Attention: Session setup failures may occur if you exhaust the entries in the NNT.

NCPTRACE
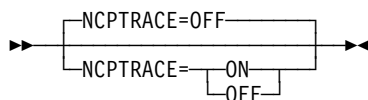
Use: 3745 frame relay and NTRI
Default: OFF

Specifies whether the snap trace function is to be enabled for 3745 frame-relay and NTRI resources.

Note: Using the snap trace degrades system performance; use it only when directed to do so by IBM support personnel.

NETID

►►──NETID=*name*──►◄

Use: Required
Default: None

Defines the name of the native network. *name* has a maximum length of 8 charac-
ters. See Table 1-4 on page 1-21 and Table 1-5 on page 1-22 for reserved pre-
fixes and labels to avoid when naming resources.

EP Standalone: NETID is not valid.

NETLIM

►►──┬──────────────┬──►◄
    └─NETLIM=*hscbs*─┘

Use: Gateway NCP
Default: See text
Maximum: 65535
Minimum: 0

Specifies how many HSCBs can be assigned at one time to all addresses in the
native network. This limit includes HSCBs in the pool for both predefined and non-
predefined resources.

Note: Even if DYNPOOL is coded, NCP will not allow the number of HSCBs to
exceed the value coded for NETLIM.

If you code SESSLIM=0 on the BUILD definition statement, the default for NETLIM
is 0 since no HSCBs can be taken from the pool. Otherwise, the default is 5000.

See NUMSESS on the GWNAU definition statement on page 2-146 for more infor-
mation on how to code NETLIM.

NEWNAME

►►──┬───────────────┬──►◄
    └─NEWNAME=*name*─┘

Use: NCP or EP
Default: See text

Defines the name of the NCP load module. NEWNAME also defines the physical
unit name for this NCP if you do not code PUNAME on the PU definition statement.
name must be a valid symbol no longer than 7 characters and must begin with an
alphabetical character or $, @, or #.

The default depends on the value of TYPGEN on the BUILD definition statement:

TYPGEN
Value   NEWNAME Default

**NCP**   NCP001

NCP-R NCP001

PEP   PEP001

| **EPSA** EP001

The name of the resource resolution table load module that corresponds to the NCP load module is the value of NEWNAME followed by the letter *R*. If a load module is generated for a block handler set resolution table, its name is the value of NEWNAME followed by the letter *B*.

Note: If a routing information table is generated for IBM Transmission Control Protocol/Internet Protocol (TCP/IP [1]), its name is the value of NEWNAME followed by the letter *P*.

NPA
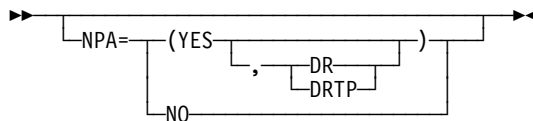
Default: See text

Specifies whether NCP is to include the network performance analyzer (NPA) function. When you code NPA=YES, NPM also collects data on the usage levels of many NCP control blocks and tables. If you code NPA=YES, define at least one virtual group with NPARSC=YES on the GROUP definition statement.

When you code SESSACC=YES on the BUILD definition statement, the default is NPA=YES; otherwise the default is NPA=NO. If you code NPA=NO, SESSACC=YES is not valid.

DR

    Specifies that NCP is to collect performance data on resources in the dynamic reconfiguration pool.

DRTP

    Specifies that NCP is to collect performance data by transmission priority in addition to the basic performance data collected on resources in the dynamic reconfiguration pool.

When GENILU=NO on the BUILD definition statement, NCP collects data for independent logical units defined with LUDR=YES only if you code NPA=(YES,DR) or NPA=(YES,DRTP).

Token Ring: Because Token-Ring logical units are allocated from the dynamic reconfiguration pool, you can code NPA=(YES,DR) to collect data for these logical units.

---

[1] For information on which product releases and operating systems are supported by NCPROUTE and IBM TCP/IP, see the TCP/IP documentation for your host operating system.

NTUNECOL

```
              ┌─NTUNECOL=NO──┐
►►────────────┴──────────────┴──►◄
         └─NTUNECOL=──┬──YES──┬───►
                      └──NO───┘
```
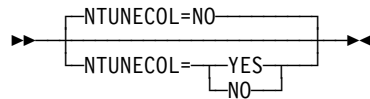
Default: NO

Specifies whether collection in the NCP is active for NTuneMON. NCP collects the following data for use by NTuneMON:

- 3745-attached SDLC lines and stations

  – Poll counts

- 3745-attached channels

  – Poll counts

- 3745 frame-relay physical stations

  – Minimum frame size transmitted
  – Maximum frame size transmitted
  – Total number of frames transmitted
  – Number of frames transmitted with BECN on (FHSPs only)
  – Number of frames transmitted with FECN on (FHSPs only)
  – Number of frames with DE on (FHSPs only)
  – Minimum frame size received
  – Maximum frame size received
  – Total number of frames received
  – Number of frames received with BECN on
  – Number of frames received with FECN on
  – Number of frames discarded

- 3745 frame-relay logical lines

  – Minimum frame size transmitted
  – Maximum frame size transmitted
  – Total number of frames transmitted
  – Number of frames with DE on
  – Minimum frame size received
  – Maximum frame size received
  – Total number of frames received
  – Number of frames received with BECN on
  – Number of frames received with FECN on
  – Number of frames discarded

- NCP Token Ring interconnection (NTRI) physical lines

  – Minimum frame size transmitted
  – Maximum frame size transmitted
  – Total number of frames transmitted

- NTRI physical stations

  – Minimum frame size transmitted
  – Maximum frame size transmitted
  – Total number of frames transmitted

- NTRI frame handler logical lines

  – Minimum frame size transmitted

- – Maximum frame size transmitted
- – Total number of frames transmitted
- • NTRI logical FHSP stations
  - – Minimum frame size transmitted
  - – Maximum frame size transmitted
  - – Total number of frames transmitted
  - – Minimum frame size received
  - – Maximum frame size received
  - – Total number of frames received
  - – Number of frames transmitted with BECN on
  - – Number of frames received with BECN on
  - – Number of frames transmitted with FECN on
  - – Number of frames received with FECN on
  - – Number of frames with DE on
  - – Number of frames discarded

Notes:

1. If NPA collection is active for a SDLC line, SDLC station, or channel, NCP collects information for these resources even if NTUNECOL=NO.

2. Specifying NTUNECOL=YES can decrease performance.

NUMHSAS

```
               ┌─NUMHSAS=128───┐
►►─────────────┤               ├──────►◄
               └─NUMHSAS=subareas─┘
```
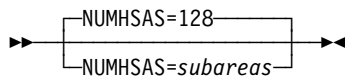
Default: 128
Minimum: 1
Maximum: See text

Specifies the expected number of subareas in the native network that have virtual routes ending in this NCP. This number gets added to the NUMHSAS values on the NETWORK definition statements. NCP does not keep a separate resource pool for each network.

In an SNA network interconnection environment, a gateway NCP can activate virtual routes. Therefore, when coding NUMHSAS, consider virtual routes originating in a gateway NCP. However, if you code an excessive value for NUMHSAS, you could waste NCP storage.

The maximum value for NUMHSAS is equal to the value of SALIMIT on this BUILD definition statement or 65534, whichever value is less. The sum of the values of NUMHSAS on the BUILD and all NETWORK definition statements must not exceed 65 534.

Although this keyword is coded for each network, the resources it creates can be used by any network within NCP. As a result, a network can have more subareas with virtual routes ending in this NCP than what it defined on its NUMHSAS keyword by using the resources defined by NUMHSAS on NETWORK definition statements for other networks.

Note: NCP can activate virtual routes for sessions between independent logical units. NCPs attached to type 2.1 nodes should be included in the number of NCP subareas with virtual routes ending in this NCP.

OLT

```
                   ┌─OLT=YES─────────┐
►►─────┬───────────────────────┬──►◄
       └─OLT=──┬─YES─┬─────────┘
               └─NO──┘
```
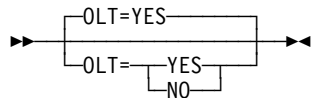
Default: YES

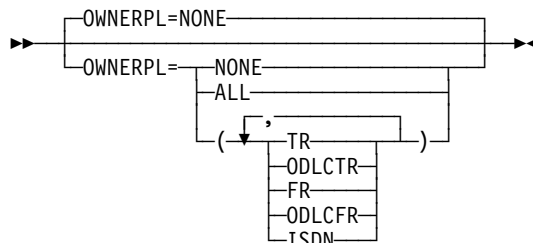Specifies whether NCP is to include the online terminal test and online line test facilities for lines operating in network control mode.

OWNERPL

```
                   ┌─OWNERPL=NONE──────────────────┐
►►─────┬──────────────────────────────────────┬──►◄
       └─OWNERPL=──┬─NONE──────────────┐
                   ├─ALL───────────────┤
                   │      ┌─────,──────┐│
                   └─(─▼─┬─TR─────┬──)─┘
                         ├─ODLCTR─┤
                         ├─FR─────┤
                         ├─ODLCFR─┤
                         └─ISDN───┘
```

Use: Token-Ring, frame-relay, and ISDN resources
Default: NONE

Specifies the types of unassigned logical resources that should be managed in owner-based pools.

An unassigned logical resource is a resource that is defined with PHYSRSC=NONE, PHYSRSC=(NONE,ODLC), or PHYSRSC=(NONE,TIC3). Without the OWNERPL function, unassigned logical resources are used for connections over any physical line of the same type. If the OWNERPL function is invoked for a particular resource type, those unassigned logical resources will be used only over a physical line (of the same type) that has the same SSCP owner as the logical line. The OWNERPL function guarantees that the logical connections over a physical line will have the same SSCP owner as the physical line.

The following types of unassigned logical resources can be specified on the OWNERPL keyword. Any or all of these values can be coded:

TR   NTRI unassigned logical peripheral resources
ODLCTR 3746 Model 900 Token-Ring unassigned logical peripheral resources
FR   3745 frame-relay unassigned logical peripheral resources
ODLCFR 3746 Model 900 frame-relay unassigned logical peripheral resources
**ISDN**   ISDN unassigned logical peripheral and subarea resources

When you code OWNERPL for a particular type of resource, you must also code the VTAM-only OWNER keyword as follows:

• Allocate each physical line of that type to a specific owner by coding OWNER, with one and only one *owner_name*, for each physical line.

- Allocate each logical pool of that type to a specific owner by coding OWNER, with one and only one *owner_name*, for each logical pool.

PATHEXT
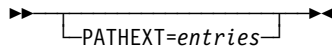
```
►►──┬───────────────────┬──►◄
    └─PATHEXT=entries─┘
```

Use: Dynamic path update
Default: See text
Minimum: 0
Maximum: 65534

Specifies how many extra transit routing table (TRT) entries for the native network are to be generated for added destination subareas.

The default is 254 minus the number of destination subareas defined for the native network.

A TRT row is generated for each unique destination subarea defined by a PATH definition statement. PATHEXT determines how many extra TRT rows and explicit-route-to-virtual-route mapping list entries are generated for the native network in addition to those generated by the PATH definition statements for the native network. The number of TRT rows generated for the native network may not exceed the value of SALIMIT on the BUILD definition statement, minus one. The number of all TRT rows in the generation may not exceed 65535.

Note: If you code an excessive value for PATHEXT, you could waste NCP storage.

PRTGEN

```
          ┌─PRTGEN=GEN─────────┐
►►──┬─┴────────────────────┴─┬──►◄
    └─PRTGEN=─┬─GEN───┬──────┘
              └─NOGEN─┘
```
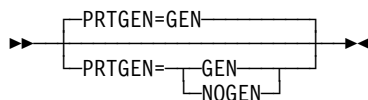
| Use: NCP or EP
        Default: GEN

Specifies whether macro-generated statements are to be printed for the NCP table assemblies.

If you code PRTGEN=GEN, the generation procedure prints all the statements generated by the phase 1 assembly. If you code PRTGEN=NOGEN, the generation procedure does not print these statements.

Note: MNOTEs are always printed, regardless of the value of PRTGEN.

PUNAME

```
►►──┬─────────────────┬──►◄
    └─PUNAME=name─┘
```
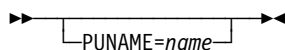
Default: None

Defines the physical unit name for this NCP. *name* must be a symbol no longer than 8 characters and must begin with an alphabetical character or $, @, or #. It

must be different from all definition statement names in the generation definition. If you omit PUNAME, NDF uses the name defined by NEWNAME on the BUILD definition statement.

PWROFF

Use: Channel-attached controller
Default: NO

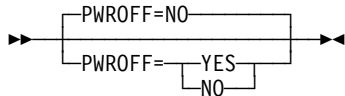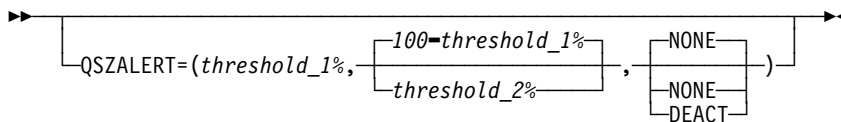Specifies whether NCP is to turn off the power in a link-attached controller upon command from a channel-attached access method. The power can be turned on again only at the control panel of the link-attached controller.

QSZALERT

Default: None
Example: QSZALERT=(50,6,NONE)

Specifies whether NCP is to check for a session or station that is using an excessive amount of buffers.

threshold_1%

    Maximum: 100 minus *threshold_2%*[2]
    Default: There is no default; if you code QSZALERT, the *threshold_1%* suboperand is required.

Specifies a minimum amount of NCP buffers that should be **available**, below which NCP checks for a session or station that is using an excessive amount of buffers. Expressed as a percentage of buffers above the slowdown entry threshold. (See SLODOWN on the BUILD definition statement on page 2-51.) Valid values are 3, 6, 12, 25, 50, 75, 87, 93, and 97. If the amount of buffers available shrinks to this percentage, NCP will alert the operator and deactivate the session or station if DEACT is coded.

threshold_2%

    Maximum: 100 minus *threshold_1%*[2]
    Default: 100 minus *threshold_1%*[2]

Specifies the maximum amount of NCP buffers that should be **in use** by a single session or station. Expressed as a percentage of buffers above the

---

[2] If necessary, subtract 1 to arrive at a valid value. See text for list of valid values.

> slowdown entry threshold. Valid values are 3, 6, 12, 25, 50, 75, 87, 93, and
> 97. If a session or station is found to be using more than the specified amount
> of buffers, the operator is alerted and, if DEACT is coded, the session or
> station is deactivated.

> **NONE** | **DEACT**

> Default: NONE

> Specifies whether NCP will deactivate the resource that is using an excessive
> amount of buffers (DEACT) or allow the resource to remain active (NONE).

REMOTTO
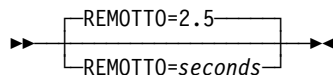
```
            ┌─REMOTTO=2.5──────┐
►►──────────┤                  ├──────►◄
            └─REMOTTO=seconds──┘
```

Use: Token-Ring logical peripheral resources and NTRI frame handler
            logical lines
Default: 2.5
Minimum: 0.6
Maximum: 20.0

Specifies, in seconds, the Token-Ring reply timer (T1) value for stations connected
to a remote Token Ring. You can specify this value as an integer or with one
decimal place. You can also specify REMOTTO using the T1TIMER keyword. See
T1TIMER on the BUILD definition statement on page 2-56 for additional informa-
tion.

RGENLEVL

```
►►──────────────────────►◄
     └─RGENLEVL=YES─┘
```

Default: See text

Specifies that NDF is to replace the GENLEVEL keyword value with the default
date and time of the NCP generation. Coding this keyword enables the date and
time of the current NCP generation to appear in the NEWDEFN file if you have
NDF create a NEWDEFN file (with NEWDEFN=YES on the OPTIONS statement).

If you do not code RGENLEVL=YES, NCP does not replace the GENLEVEL
keyword value.

SALIMIT

```
            ┌─SALIMIT=255─────────┐
►►──────────┤                     ├──────►◄
            └─SALIMIT=──┬─255───┬─┘
                        ├─511───┤
                        ├─1023──┤
                        ├─2047──┤
                        ├─4095──┤
                        ├─8191──┤
                        ├─16383─┤
                        ├─32767─┤
                        └─65535─┘
```
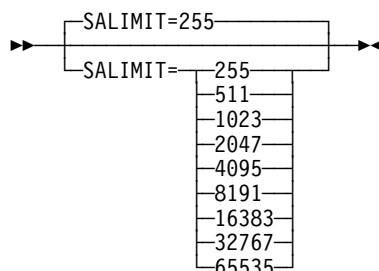
Default: 255
Minimum: 255
Maximum: 65535

Specifies the largest subarea address supported by this node in the native network.

SESSACC

```
►►─┬─SESSACC=NO────────────────────────────────────────────┬─►◄
   │                                                         │
   └─SESSACC=─┬─(YES,─┤ Accounting parameters ├─)─┬─────────┘
             └─NO───────────────────────────────┘
```

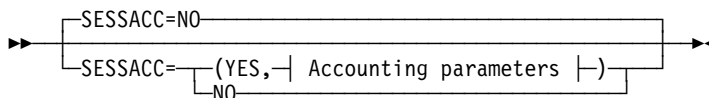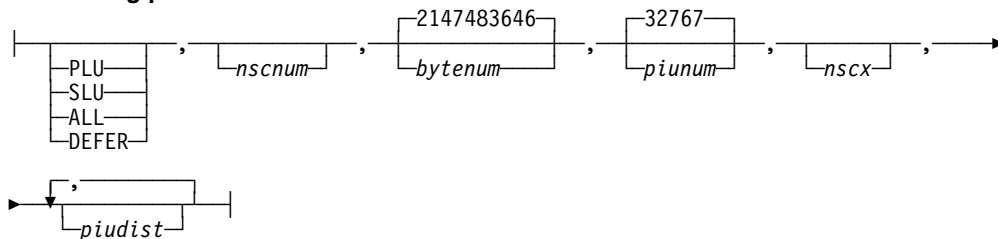**Accounting parameters:**

```
├─┬───────┬─,─┬────────┬─,─┬──2147483646──┬─,─┬──32767──┬─,─┬──────┬─,─►
  ├─PLU───┤   └─nscnum─┘   └──bytenum─────┘   └─piunum──┘   └─nscx─┘
  ├─SLU───┤
  ├─ALL───┤
  └─DEFER─┘

►─┬──────────────┬─┤
  │   ┌──,──┐     │
  └─▼─┴─piudist─┴─┘
```

Use: Required for session accounting
Default: NO
Example: SESSACC=(YES,,,65535,65535)

Specifies whether NCP is to include session accounting. SESSACC=YES is not valid if you code NPA=NO on the BUILD definition statement.

PLU|SLU|ALL|DEFER

Default: None

Specifies that NCP is to perform session accounting for the primary logical unit (PLU) session block, the secondary logical unit (SLU) boundary session block, or both (ALL), or that data collection is to be deferred until NCP receives a change-accounting PIU (DEFER).

nscnum

Default: See text
Minimum: 0
Maximum: See text

Specifies how many NPA session accounting blocks (NSCs) are to be generated for session accounting.

When you code the PLU suboperand, the default is the number of independent boundary session blocks (BSBs). Otherwise, the default is the number of BSBs plus the number of dependent logical unit control blocks (LNDs).

The maximum is the number of BSBs plus the number of LNDs.

If the total number of NSCs defined by the SESSACC and GWSESAC keywords exceeds 65 535, NDF generates 65 535 NSCs.

Note: You can define part of the NCP buffer pool to be used for dynamically created NSCs when explicitly defined NSCs have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on

page 2-19. See DYNNSC on the BUILD definition statement on page 2-18 for information about defining limits within dynamically created control block pools and see the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

bytenum

Default: 2147483646 ($2^{31}-2$)
Minimum: 1
Maximum: 4294967295 ($2^{32}-1$)

Specifies how many bytes NCP is to count before sending session accounting data to NPM. *bytenum* must be greater than or equal to *piunum*.

Note: The value 2147483648 (2^{31}) is not valid.

piunum

Default: 32767
Minimum: 1
Maximum: 65535

Specifies how many PIUs NCP is to count before sending session accounting data to NPM. *piunum* must be less than or equal to *bytenum*.

If you use boundary function session accounting, use *bytenum* and *piunum* values appropriate for your own configuration. Do not use the default values.

Note: Do not make *bytenum* and *piunum* too small. Small values degrade NCP performance because too much time is spent sending data to NPM.

nscx

Default: See text
Minimum: 0
Maximum: 65535

Specifies how many NSC extensions are to be generated. The default is 3 or 45% of *nscnum*, whichever is greater.

If the total number of NSC extensions defined by the SESSACC and GWSESAC keywords exceeds 65 535, NDF generates 65 535 NSC extensions.

When session accounting records are delayed, NSC extensions are needed to save information that would otherwise be lost. Up to three NSC extensions may be needed for each session for which session accounting information is delayed.

Note: You can define part of the NCP buffer pool to be used for dynamically created NSC extensions when explicitly defined NSC extensions have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See DYNNSX on the BUILD definition statement on page 2-18 for information about defining limits within dynamically created control block pools and see the *NCP, SSP,*

and *EP Resource Definition Guide* for more information about dynamically created control blocks.

piudist

Default: None
Minimum: 0
Maximum: 65535

Specifies up to six PIU distribution lengths by which PIUs flowing through a particular session are to be counted. Code *piudist* values in ascending order.

Code at least one *piudist* value if you want to be able to specify PIU distribution ranges at a later time. If you do not define any PIU distribution ranges, NCP does not include storage for the distribution counters.

If you code a *piudist* value of 0, all subsequent *piudist* values are ignored.

SESSLIM

```
                ┌─SESSLIM=255──┐
►►──────────────┼──────────────┼──────►◄
                └─SESSLIM=hscbs─┘
```
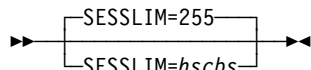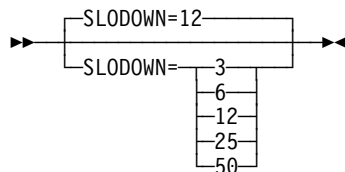
Use: Gateway NCP
Default: 255
Minimum: 0
Maximum: 255

Specifies how many HSCBs can be assigned to an address in the native network. This keyword is valid only if you define HSBPOOL.

Remember that although an address resides in the native network, the SSCP or logical unit that it represents resides in a non-native network.

In some cases, SESSLIM also specifies the maximum number of sessions in which predefined resources can participate. See NUMSESS on the GWNAU definition statement on page 2-146 for more information.

SLODOWN

```
         ┌─SLODOWN=12──────┐
►►───────┼─────────────────┼──►◄
         └─SLODOWN=─┬─3──┬─┘
                    ├─6──┤
                    ├─12─┤
                    ├─25─┤
                    └─50─┘
```

Default: 12

Defines the NCP slowdown threshold as a percentage of the total number of NCP buffers. NCP enters slowdown mode when the number of available buffers drops below this percentage.
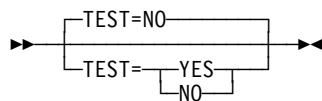
SUBAREA

```
►►──SUBAREA=address──►◄
```

Use: Required
Default: None
Minimum: 1
Maximum: See text

Assigns a subarea address to this NCP or PEP. The largest address you can code
is the value of SALIMIT on the BUILD definition statement.

EP Standalone: SUBAREA is not valid.

TEST

```
        ┌─TEST=NO─────────┐
►►──────┤                 ├──►◄
        └─TEST=──┬─YES─┬──┘
                 └─NO──┘
```
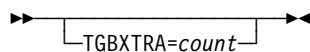
Use: Emulation mode
Default: NO

Specifies whether the Emulation Program is to include the line test function, which
is initiated from the maintenance and operator subsystem (MOSS) console of the
communication controller.

For more information about the line test function, see the *IBM 3745 Problem Deter-
mination Guide*.

TGBXTRA

```
►►──┬──────────────────┬──►◄
    └─TGBXTRA=count─────┘
```

Use: Dynamic path update
Default: See text
Minimum: 0
Maximum: 65534

Specifies how many extra transmission group control blocks (TGBs) are to gener-
ated in addition to those generated by the PATH definition statements for the native
network.

Each extra TGB allows one new adjacent subarea and transmission group number
(*adjsa,tgn*) pair to be defined in NCP by a dynamic path update. You specify the
adjsa,tgn pair on the ER0 through ER15 keywords on the PATH definition state-
ment. A new *adjsa,tgn* pair is one that was not specified on a PATH definition
statement in the NCP generation definition, but is on one or more PATH definition
statements in the dynamic path update members. The dynamic path update
members are used to change the paths of this NCP.

The total number of TGBs generated by the BUILD and NETWORK definition state-
ments must not exceed 65 534.

The default for TGBXTRA is the total number of subarea links and subarea channels in this NCP minus the total number of *adjsa,tgn* pairs defined by all PATH definition statements.

Note: If you code an excessive value for TGBXTRA, you could waste NCP storage.

TRACE

```
                        ┌─TRACE=NO──────────────────────┐
►►──────────────────────┼───────────────────────────────┼──►◄
                        │              ┌─50─┐            │
                        └─TRACE=──┬─(YES,─┼────┼─────)─┘ │
                                  │           └─entries─┘
                                  └─NO──────────────────────┘
```
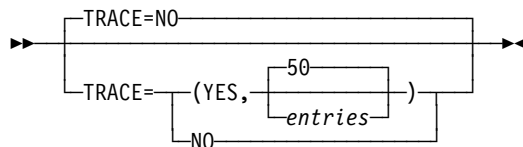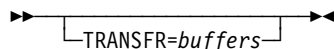
Default: NO

Specifies whether NCP is to include the address trace.

entries

Default: 50
Minimum: 10
Maximum: 512

Specifies how many 16-byte entries the trace table is to contain.

TRANSFR

```
►►───┬──────────────────┬───►◄
     └─TRANSFR=buffers─┘
```

Default: See text
Minimum: See text
Maximum: 254

Defines the default value for TRANSFR on the LINE definition statement. See TRANSFR on the LINE definition statement on page 2-286 for more information.

The value of TRANSFR on the BUILD definition statement times the value of BFRS on the BUILD definition statement must be greater than 1296.

If you do not inhibit subblocking, the minimum value for TRANSFR is 2.

The default is the number of NCP buffers required to hold a 4096-byte PIU. To calculate this value, add 51 to 4096, divide the result by the value of BFRS, and round to the next highest integer.

If you use the VTAM MODIFY LOAD command to add or replace a load module on the communication controller hard disk, the TRANSFR value times the BFRS value must be at least 2048.

Type 7 Channel Adapter: Specify a value as close as possible to the size of the largest PIU NCP will receive. Specifying too large a value can degrade performance and waste NCP storage.

TRCPIU

```
             ┌─TRCPIU=2000─┐
►►───────────┤             ├───►◄
             └─TRCPIU=bytes─┘
```
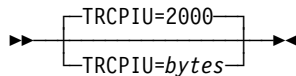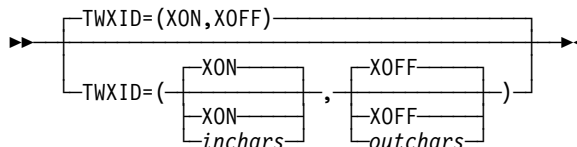
Default: 2000
Minimum: See text
Maximum: See text

Specifies, in bytes, the size of the line trace and scanner interface trace PIUs. The minimum value is two times the NCP buffer size (as defined by BFRS on the BUILD definition statement) plus 40. For example, if the NCP buffer size is 200 bytes, the minimum for TRCPIU is $(2 \times 200) + 40 = 440$. The maximum value is 255 times the buffer size.

TRCPIU limits the amount of line trace data accumulated in NCP buffers to no more than the access method (and intermediate access methods, if any) in the destination host processor can accept in a single PIU. TRCPIU also limits the amount of line trace data to no more than the data transfer limit imposed by a subsequent NCP in the path.

3746 Model 900 Token Ring: When you specify NCP support for the 3746 Model 900 connectivity subsystem on the VERSION keyword (VERSION=V*x*R*y*F), the minimum value for TRCPIU is 550.

TWXID

```
             ┌─TWXID=(XON,XOFF)──────────────────────┐
►►───────────┤                                       ├───►◄
             │            ┌─XON──┐      ┌─XOFF──┐     │
             └─TWXID=(────┼─XON──┼──,───┼─XOFF──┼──)──┘
                          └─inchars─┘   └─outchars─┘
```

Use: TWX terminals on switched lines in network control mode

Specifies the characters used for both the identifier answerback sequence (or identifier) and the prompting sequence (or PROMPT) sent as part of each TWX terminal READ operation.

inchars

> Default: XON
>
> Specifies the answerback and prompting sequences sent to all TWX terminals when the switched connection is initiated by the terminal.

outchars

> Default: XOFF
>
> Specifies the answerback and prompting sequences sent to all TWX terminals when the switched connection is initiated by the controller.

Specify *inchars* and *outchars* as the hexadecimal representation of the EBCDIC characters. Each can contain up to 20 characters. You can specify any EBCDIC character, including those that translate into TWX carriage return or line feed.

If an initial identifier is to be sent, you must code XTWXID=YES on the TERMINAL definition statement. If you want to use prompting, then you must code PROMPT=YES on the LINE definition statement. Identifiers normally end with a WRU character. Prompts *always* end with an XON character.

Because the prompting sequence is sent at the beginning of every terminal read operation, specify a short sequence to avoid using an excessive amount of time when transmitting and printing the prompting sequence.

Attention: Because TWXID is used to define both the identifier and prompting sequences, be sure you understand that when you code TWXID for the identifier, NCP overlays the last character of the TWXID sequence with an XON character when it transmits a prompting sequence. If you are not using identifiers, code TWXID for the prompting sequence (for example, the last character is an XON character).

TYPGEN

```
►►──TYPGEN=──┬─NCP───┬──►◄
             ├─NCP-R─┤
             ├─PEP───┤
             └─EP────┘
```
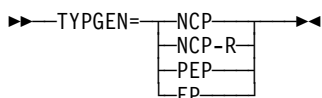
Use: Required for NCP or PEP; optional for EPSA
Default: NCP or PEP — None.
            EPSA — The default is TYPGEN=EP.

Specifies (1) whether the program you are defining will operate the communication controller as channel-attached or link-attached, and (2) whether the program includes NCP, NCP and EP or only EP functions. Select the appropriate value from Table 2-8.

Table 2-8. Values for TYPGEN on the BUILD Definition Statement

Controller Operation	EP Functions	TYPGEN Value
Channel-attached and link-attached	No	NCP
Link-attached	No	NCP-R
Channel-attached and link-attached	Yes	PEP
EP standalone	Yes	EP

TYPSYS
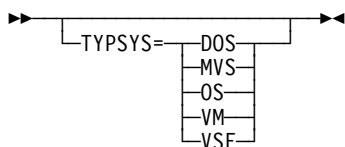
```
►►──┬────────────────────────┬──►◄
    └─TYPSYS=──┬─DOS─┬────────┘
               ├─MVS─┤
               ├─OS──┤
               ├─VM──┤
               └─VSE─┘
```
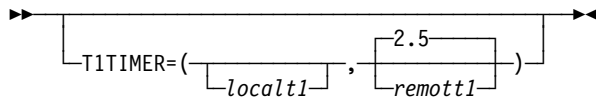
Use: NCP or EP
Default: See text

Specifies whether this NCP will be generated under an MVS (TYPSYS=MVS or TYPSYS=OS), VM (TYPSYS=VM), or VSE (TYPSYS=VSE or TYPSYS=DOS) operating system. The default for TYPSYS is the operating system used for the gen.

T1TIMER

```
►►─┬──────────────────────────────────────────┬─►◄
   │                            ┌──2.5──┐       │
   └─T1TIMER=(─┬─────────┬─,─┬─────────┬─)─┘
               └─localt1─┘     └─remott1─┘
```

Use: Token-Ring logical peripheral resources and NTRI frame handler logical lines

Specifies, in seconds, the Token-Ring reply timer (T1) values for stations connected to a Token-Ring.

localt1

Default: See text
Minimum: 0.6
Maximum: 20.0

Specifies the reply timer value for stations connected to a local Token Ring. The value must be less than or equal to the value of *remott1*. *localt1* is used for all peripheral connections and any subarea connections that do not have a different value coded on the GROUP, LINE, or PU definition statement.
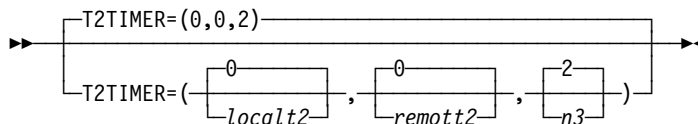
The default for *localt1* is either 1.5 or the value of *remott1*, whichever is less.

remott1

Default: 2.5
Minimum: 0.6
Maximum: 20.0

Specifies, in seconds, the reply timer (T1) value for stations connected to a remote Token Ring. You can specify this value as an integer or with one decimal place.

T2TIMER

```
          ┌──T2TIMER=(0,0,2)──────────────────────────┐
►►─┼───────────────────────────────────────────────┼─►◄
   │              ┌──0──┐       ┌──0──┐     ┌──2──┐   │
   └─T2TIMER=(─┬─────────┬─,─┬─────────┬─,─┬─────┬─)─┘
               └─localt2─┘     └─remott2─┘   └─n3──┘
```

Use: Token-Ring logical peripheral resources

Specifies the Token-Ring acknowledgment timer (T2) values and the maximum number of I-frames NCP can receive without sending an acknowledgment.

If possible, consider the value of the reply timer (T1) used by the adjacent station when selecting values for the acknowledgment timer. The acknowledgment timer interval should be about one-tenth as long as the reply timer interval for the adjacent station to ensure that the adjacent station receives the delayed acknowledgment before its reply timer expires.

localt2

Default: 0
Minimum: 0
Maximum: 2

Specifies the timer value for connections over a local Token Ring. This value is used for all peripheral connections and any subarea connections that do not have a different value coded on the GROUP, LINE, or PU definition statement. You can specify this value as an integer or with one decimal place. Code a value of 0 if you do not want to use this timer.

Frame Relay: The *localt2* value is not used as a default for frame-relay lines. See T2TIMER on the LINE definition statement on page 2-291 for more information about coding an acknowledgment timer value for frame-relay lines.

remott2

Default: 0
Minimum: 0
Maximum: 2

Specifies the timer value for connections over a remote Token Ring. This value is used for all peripheral connections and any subarea connections that do not have a different value coded on the GROUP, LINE, or PU definition statement. You can specify this value as an integer or with one decimal place. Code a value of 0 if you do not want to use this timer.

n3

Default: 2
Minimum: 1
Maximum: 127

Specifies the maximum number of I-frames NCP is to receive before sending an acknowledgment. This value is used only for subarea lines that do not have a different value coded on the GROUP or LINE definition statement.

n3 is not used if the value of *localt2* and *remott2* are defaulted or coded with a value of 0, and acknowledgments will be sent for each received I-frame.

For peripheral connections, a value of 2 for *n3* is used if nonzero values are coded for *localt2* or *remott2*.

UCHAN

```
                   ┌─UCHAN=NO─────────┐
►►─────────────────┴─UCHAN=──┬─YES─┬──┴─────────►◄
                             └─NO──┘
```
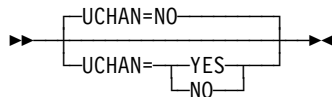
Default: NO

Specifies whether user-written channel handling code is provided for channel adapter logical addresses as follows:

Controller Model	CA Addresses
IBM 3745-130, 3745-170, or 3745-17A	0 to 3
IBM 3745-x10 or 3745-x1A	0 to 15

If you specify UCHAN=YES, the user-written channel handling code must be assembled in phase 2 of the generation through the SRCLO keyword on the GENEND definition statement or by using the NDF standard attachment facility. You also need to do the following:

* Code the HICHAN and LOCHAN keywords on the LINE definition statement to specify the highest and lowest subchannel addresses that can be used by either the user-written channel code or any EP lines.

* Code CHANLNK=YES on at least one GROUP definition statement.

To define subchannels for user-written channel code, you need to code at least two GROUP definition statements:
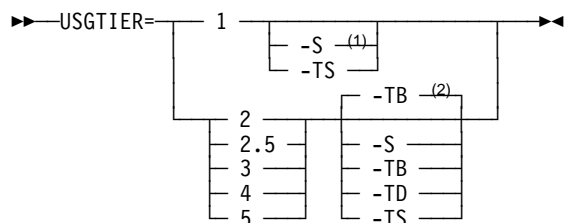
1. Code one GROUP definition statement with CHANLNK=YES and LNCTL=SDLC. Code ADDRESS on the LINE definition statement to define the channel adapter logical address. Code LOCADDR on the LU definition statement to define the user subchannels. The subchannel addresses must be within the range specified for this channel adapter by HICHAN and LOCHAN.

2. Code another GROUP definition statement with LNCTL=CA. Code ADDRESS on the LINE definition statement to define the channel adapter logical address. This address should match an address on a GROUP definition statement with CHANLNK=YES. Code HICHAN and LOCHAN on the LINE definition statement with a range that includes the subchannel addresses defined by LOCADDR on the LU definition statement. These subchannel addresses are for the logical units on the lines under the GROUP definition statement with CHANLNK=YES.

Table 2-9 shows the correlation of channel link address to subchannel address for user subchannels.

Table 2-9. Correlation of Channel Link Address to Subchannel Address for User-Written Subchannels

Channel Link Address		**Subchannel Address**	
GROUP	LNCTL=CA	GROUP	LNCTL=SDLC
			CHANLNK=YES
LINE	ADDRESS=0	LINE	ADDRESS=0
	HICHAN=07	LU	LOCADDR=7
	LOCHAN=00		

USGTIER

```
►►──USGTIER=──┬─ 1 ──────────────────────────────►◄
              │      ┌─ -S ─(1)─┐
              │      ├─ -TS ────┘
              │              ┌─ -TB ─(2)─┐
              ├─ 2 ──────────┤           │
              ├─ 2.5 ────────┼─ -S ──────┤
              ├─ 3 ──────────┼─ -TB ─────┤
              ├─ 4 ──────────┼─ -TD ─────┤
              └─ 5 ──────────┴─ -TS ─────┘
```

[1] For usage tier 1, the -S suffix is not valid for 3745-410/41A/610/61A.

[2] Defaults for usage tiers 2 through 5:

- -TB is the default for tiers 2 and 2.5, but only if MODEL=3745-41A or 3745-61A. Otherwise there is no default for tiers 2 and 2.5.

- -TB is the default for tier 3.

- -TB is not a default for usage tiers 4 and 5. There is no default for usage tiers 4 and 5.

Use: Required
Default: See the second note under the USGTIER syntax diagram.

Specifies the usage tier for this NCP. The usage tier determines the number of scanners, Token-Ring adapters, Ethernet-type subsystem (ESS) adapters, and channel adapters you can attach to your controller. To determine the usage tier level you require, look at your IBM Customer Agreement. See ADDRESS on the LINE definition statement on page 2-180 for valid addresses for each usage tier. See the *NCP, SSP, and EP Resource Definition Guide* for a general discussion of usage tiers.

If you specify a USGTIER value that exceeds the usage tier installed with your NCP, NCP generation will fail and addresses will not be checked. USGTIER=4 is not valid if you code MODEL=3745-160. USGTIER=5 is not valid if you code MODEL=3745-130, MODEL=3745-150, MODEL=3745-160, MODEL=3745-170, or MODEL=3745-17A.

Acceptable airline line control integration (ALCI) usage tiers for each controller are:

- Usage tier 2 or higher for the IBM 3745-130, 3745-150, 3745-160, or 3745-170
- Usage tier 3 or higher for the IBM 3745-210, 3745-310, 3745-410, and 3745-610

Table 2-10 shows valid operating modes for the following 3745 communication controller models and the default (if any) when you code a usage tier level without a suffix. Models not listed operate only in Single mode.

Table 2-10. Valid Usage Tier Suffix Values			
When You Code MODEL=	**And Usage Tier Level=**	**Default Suffix**[1]	**Valid Suffix Values**[1]
3745	1, 2, or 2.5	none	S, TS
	3	TB	S, TD, TS, TB[2]
	4 or 5	none	
3745-410 3745-41A 3745-610 3745-61A	1	none	TS
3745-410 3745-610	2 or 2.5	none	TS
	3	TB	TS, TD, TB[2]
	4 or 5	none	
3745-41A 3745-61A	2, 2.5, or 3	TB	TD, TS, TB[2]
	4 or 5	none	
[1] Suffix meanings: -S Single mode -TS Twin-in-standby mode -TD Twin-in-dual mode -TB Twin-in-backup mode [2] You can use -TB or -TD regardless of the controller model or its actual operating mode.			

NDF uses the suffix you code to validate the number of adapters defined in your generation definition. When your controller is operating in single or twin-in-standby mode, NCP can access all the adapter positions. Consequently, when you code -S or -TS, you can define as many adapters as the usage tier allows for your controller model. When your controller is operating in twin-in-dual or twin-in-backup mode, NCP can access only half of the adapter positions. Consequently, when you code -TD or -TB, you can define only half as many adapters as the usage tier allows for your controller model. Do not load an NCP generated with the -S or -TS suffix into an IBM 3745 actually operating in twin-in-dual or twin-in-backup mode because NDF does not verify that the adapters defined are valid for these operating modes.

For more information on the number and positions of adapters allowed for the various controller models and usage tiers, see Table 2-51 on page 2-185. For more information on the different twin-CCU operating modes, see Chapter 6, "Support for the IBM 3745 Communication Controller," in the NCP and EP Reference manual.

Note: The USGTIER keyword applies only to 3745 adapters. See CSSTIER on the BUILD definition statement on page 2-15 for information about specifying usage tiers for the 3746 Model 900.

EP Standalone: USGTIER is not valid.

VERSION

```
►►──VERSION=──┬─VxRy──┬──────►◄
              └─VxRyF─┘
```

| Use: | Required (NCP or EP) |
| Default: | None |

Specifies the NCP or EP version (V*x*) and release (R*y*) to be generated.

3746 Model 900 CSS:

1. To specify that you want NCP support for the 3746 Model 900 connectivity sub-system, use the V*x*R*y*F suboperand. (V*x*R*y*F is valid only for V6R2 or later.)

2. Please note that you must obtain a license for the 3746 Model 900 CSS feature. Refer to Order instructions in the NCP announcement letter for ordering information.

EP Standalone: Code VERSION=V1R14.

VOSDEACT

```
            ┌─VOSDEACT=NO────────────┐
►►──────────┴─VOSDEACT=──┬─YES─┬──────┴──────►◄
                         └─NO──┘
```

Default: NO

VOSDEACT=YES specifies that VR deactivation will be initiated when a virtual route (VR) goes out-of-sequence.

VOSDEACT=NO specifies that VR deactivation will not be initiated when a VR goes out-of-sequence.

For both VOSDEACT=NO and VOSDEACT=YES, the network operator is alerted upon detection of the out-of-sequence VR.

VRPOOL

```
►►──┬──────────────────────────────────┬──►◄
    └─VRPOOL=(──┬─────────┬─┬───────┬─)─┘
               └─entries─┘ └─,rows─┘
```

Defines the number of entries in the virtual route pool for this NCP and specifies how many extra rows the flow control parameter table is to contain.

entries

Default: See text
Minimum: 1
Maximum: 65532

Defines the number of entries in the virtual route pool for this NCP. The number of entries is determined by adding the following:

- The number of concurrently active virtual routes with an end point in this NCP

- The maximum number of virtual routes that can be deactivated and then reactivated because of an explicit route failure

- The number of active internal virtual routes needed by user-written code

The default is 768 times the number of NETWORK definition statements in the generation definition. If there are no NETWORK definition statements, the default is 768.

rows

Default: Value of *entries*
Minimum: 0
Maximum: Value of *entries*

Specifies how many extra rows the flow control parameter table is to contain.

Note: If you code excessive values for VRPOOL, you could waste NCP storage.
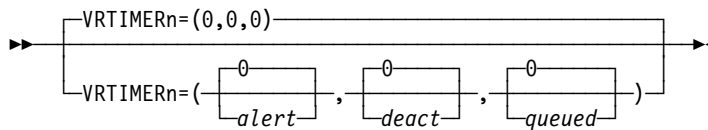
VRTIMERn

```
     ┌─VRTIMERn=(0,0,0)────────────────────┐
►►───┤                                     ├──►◄
     │              ┌─0─┐    ┌─0─┐    ┌─0────┐ │
     └─VRTIMERn=(───┤   ├──,─┤   ├──,─┤      ├─)
                    └─alert─┘ └─deact─┘ └─queued─┘
```

Specifies the conditions under which NCP is to send a blocked virtual route alert to the NetView program or deactivate a held virtual route. For more information about blocked virtual route alerts, see the *NCP and EP Reference*.

In VRTIMER*n*, *n* can be 0, 1, or 2. VRTIMER0, VRTIMER1, and VRTIMER2 indicate timer values applied to virtual routes with a transmission priority of 0, 1, and 2, respectively.

alert

Default: 0
Minimum: 0
Maximum: 86400

Specifies, in seconds, the virtual route congestion limit. If a virtual route is congested for this length of time, NCP sends a blocked virtual route alert to the NetView program. NCP sends the alert even if the virtual route pacing response is received during the time the virtual route is congested. If the value of *alert* is 0, NCP does not send blocked virtual route alerts.

If the value you code is not an even multiple of 10, NDF rounds the value up to the nearest multiple of 10. The maximum of 86400 seconds equals one day.

deact

Default: 0
Minimum: 0
Maximum: 86400

Specifies, in seconds, the held virtual route limit. If a virtual route is held for this length of time, NCP deactivates the route and sends an alert to the NetView program. If the value of *deact* is 0, NCP does not deactivate virtual routes because of a held condition.

If you code a value other than 0 for *deact*, that value must be greater than or equal to *alert*. If the value you code is not an even multiple of 10, NDF rounds the value up to the nearest multiple of 10. The maximum of 86400 seconds equals one day.

queued

Default: 0
Minimum: 0
Maximum: 255

Specifies the number of PIUs that constitutes congestion. If the number of PIUs on a virtual route's transmit queue is equal to or greater than *queued*, the virtual route is congested. If the value of *queued* is 0, NCP does not send alerts for congested virtual routes.

If you code 0 for *alert*, you must also code 0 for *queued*.

VSUBNETS

```
                     ┌─VSUBNETS=NO──────────┐
►►─┬──────────────────────────────────┬─►◄
   └─VSUBNETS=─┬─YES─┬────────────────┘
              └─NO──┘
```
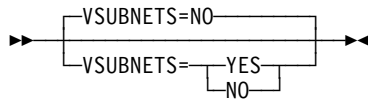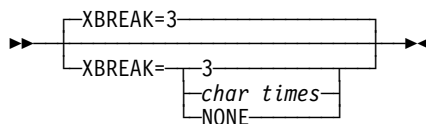
Use: Internet Protocol subnetworks
Default: NO

Specifies whether variable subnetting is supported via NDF and dynamic reconfig-
uration. VSUBNETS=YES specifies that IP interfaces in, and IP routes to, the
same IP network can be defined with different subnet masks. VSUBNETS=NO
enforces NDF checking of, and dynamic-reconfiguration processing of, IP interfaces
and routes such that routes requiring variable subnetting support are flagged as in
error.

Notes:

1. VSUBNETS=NO does not prevent NCPROUTE from adding variable-subnetted
   routes when NCPROUTE is configured to use RIP Version 2 (RIP-2).

2. NDF does not provide a SNETMASK keyword default value for any IP interface
   definition when you specify VSUBNETS=YES. For additional information, see
   SNETMASK on the PU definition statement on page 2-415.

3. VSUBNETS=YES must be specified to enable receiving Ethernet multicast
   frames for RIP-2. The Ethernet adapter (ESS) patch EGAL2 must also be
   applied when VSUBNETS=YES.

XBREAK

```
                 ┌─XBREAK=3────────────┐
►►─┬──────────────────────────────┬─►◄
   └─XBREAK=─┬─3──────────┬────────┘
            ├─char times─┤
            └─NONE───────┘
```

Use: Start-stop lines in network control mode
Default: 3

Specifies the length of the break signal NCP is to place on lines to interrupt trans-
missions from terminals. This keyword applies only to terminals for which all of the
following are true:

- The line is in network control mode.
- DUPLEX=FULL on the LINE definition statement.
- FEATURE=BREAK on the TERMINAL definition statement.

char times
    Specifies the length, in character times (the time required to transmit one char-
    acter), of the break signal.

NONE
    Specifies that NCP is not to send break characters.

CLUSTER Definition Statement

Usage: Clustered BSC stations. One for each cluster controller.

Position: Must follow SERVICE definition statement if SERVICE is coded. If SERVICE is not coded, must follow the LINE definition statement. Must precede other device definition statements (TERMINAL and COMP) for the cluster.

Keywords: Table 2-11 shows the keywords described under this definition statement.

Table 2-11. Keywords on the CLUSTER Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
BHEXEC	2-67		G,L		
BHSET	2-68		G,L	NONE	
CUTYPE	2-68		G,L	3271	
EXEC	2-68		G,L	YES	
FEATURE	2-69		G,L	NOGPLKUP	
GPOLL	2-69				
INHIBIT	2-70		G,L	NONE	
ITBMODE	2-71		G,L		
LGRAPHS	2-71		G,L	(REJECT,REJECT)	
LMODADR	2-72		PU		
NEXT	2-72				
NPACOLL	2-73		G,L	NO	
PT3EXEC	2-74		G,L	NO	
SEGMENT	2-74		PU		

Sift:

G   GROUP definition statement
L   LINE definition statement
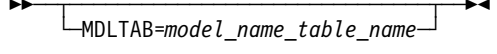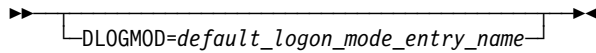PU   PU definition statement
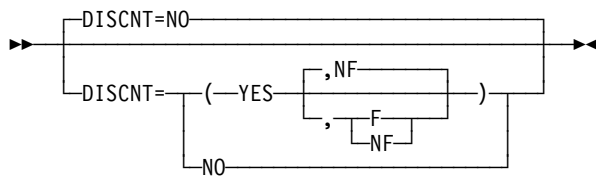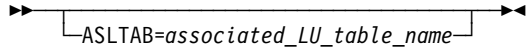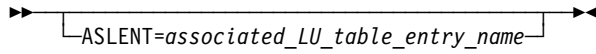LU   LU definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements: In addition to the keywords described in this section, you can define most keywords of the TERMINAL definition statement on the CLUSTER (for cluster-type stations), LINE, or GROUP definition statement. Table 1-3 on page 1-12 shows which of the keywords you can code on CLUSTER or at a higher level.

VTAM Keywords: The following diagrams show the VTAM keywords that can be coded on this definition statement. These keywords provide information to the VTAM initialization process. They are not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for descriptions of these keywords and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

| VTAM keywords not listed below can be specified by coding VTM. as a prefix in
| front of the VTAM keyword.

```
►►─┬──────────────────────────────────────────────┬─►◄
    └─ASLENT=associated_LU_table_entry_name─┘
```

```
►►─┬────────────────────────────────────┬─►◄
    └─ASLTAB=associated_LU_table_name─┘
```

```
        ┌─DISCNT=NO────────────────────────────────┐
►►─┼──────────────────────────────────────────┼─►◄
    └─DISCNT=─┬─(──YES─┬──────┬────────┬─)─┘
              │        │ ┌─,NF─┐ │        │
              │        └─,─┤    ├─┘        │
              │            │ ,─┬─F──┐ │
              │            └───┤NF─┘ │
              └─NO────────────────────────┘
```

```
►►─┬──────────────────────────────────────────────┬─►◄
    └─DLOGMOD=default_logon_mode_entry_name─┘
```

```
►►─┬───────────────────────────┬─►◄
    │            ┌─,───────┐    │
    └─FEATUR2=(──▼─keyword─┴──)─┘
```

```
        ┌─ISTATUS=ACTIVE──────┐
►►─┼─────────────────────┼─►◄
    └─ISTATUS=─┬─ACTIVE───┬─┘
              └─INACTIVE─┘
```

```
►►─┬─────────────────────────────────────┬─►◄
    └─LOGAPPL=application_program_name─┘
```

```
►►─┬────────────────────────────────┬─►◄
    └─LOGTAB=interpret_table_name─┘
```

```
►►─┬──────────────────────────────────────┬─►◄
    └─MDLENT=model_name_table_entry_name─┘
```

```
►►─┬───────────────────────────────┬─►◄
    └─MDLTAB=model_name_table_name─┘
```

```
►►─┬──────────────────────────────┬─►◄
    └─MODETAB=logon_mode_table_name─┘
```

```
►►─┬─────────────────────────────────┬─►◄
    └─USSTAB=USS_definition_table_name─┘
```

```
        ┌─VPACING=2─┐
►►─┼───────────┼─►◄
    └─VPACING=n─┘
```
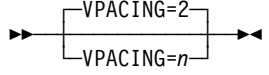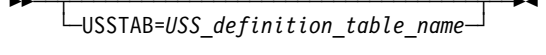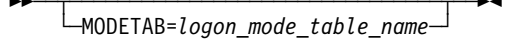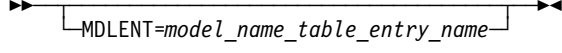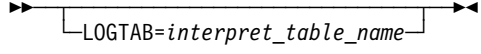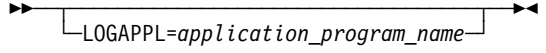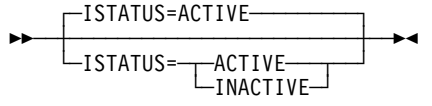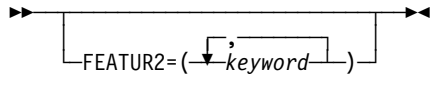
Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the CLUSTER definition statement
Default: None

Provides a name for the station. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

symbol is a resource name for the cluster only if you code GPOLL on this definition statement. The symbol appears in the label cross-reference listing that the generation process creates. If AUTOLINE is specified, you must embed the first suboperand of the ADDRESS keyword on the LINE definition statement in the symbol.

BHEXEC
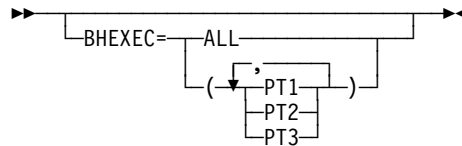
Use: Block handling routines
Default: See text

Specifies which block handling routines NCP is to use from the block handler set defined by the BHSET keyword. If you code BHSET=*setname*, specify at least one routine for BHEXEC. Do not code BHEXEC if you specify BHSET=NONE or BHSET=DYNAMIC. If you omit BHEXEC and define BHSET, ALL is the default.

ALL
   Specifies that each block handler in the set is used at the appropriate time.

PT1
   Specifies that NCP uses the PT1 block handler. NCP calls this block handler upon receiving a request from the host processor but before determining whether the line is available to contact the station.
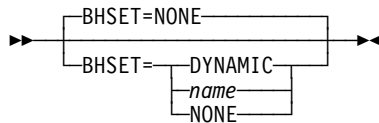
PT2
   Specifies that NCP uses the PT2 block handler. NCP calls this block handler upon receiving a request from the host processor and after determining that the line is available.

PT3
   Specifies that NCP uses the PT3 block handler. NCP calls this block handler after receiving a block, message, or transmission from the station.

BHSET

```
                 ┌─BHSET=NONE─────────┐
►►───────────────┴────────────────────┴──►◄
          └─BHSET=──┬─DYNAMIC─┬─┘
                    ├─name────┤
                    └─NONE────┘
```

Use: Block handling routines
Default: NONE

Specifies the name of a set of block handlers associated with this station.

For BSC 3270 devices, VTAM ignores BHSET.

NONE
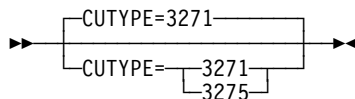> Specifies that no block handler set is assigned to this station.

DYNAMIC
> Specifies that no block handler set is assigned initially to this station, but one can be assigned dynamically from the host processor.

name
> Specifies the block handler set for this station. *name* must be the name of a BHSET definition statement.

CUTYPE

```
             ┌─CUTYPE=3271─────┐
►►───────────┴─────────────────┴──►◄
       └─CUTYPE=──┬─3271─┬─┘
                  └─3275─┘
```

Default: 3271

Specifies whether the control unit for this station is an IBM 3271 or 3275 control unit. Code CUTYPE=3271 or omit this keyword if the control unit is an IBM 3274 or 3276 operating in BSC mode. If the line is connected to IBM 3270 stations and operates in emulation mode as well as in network control mode, specify CUTYPE on both the LINE and CLUSTER definition statements.
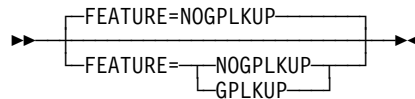
EXEC

```
             ┌─EXEC=YES────┐
►►───────────┴─────────────┴──►◄
       └─EXEC=──┬─YES─┬─┘
                └─NO──┘
```
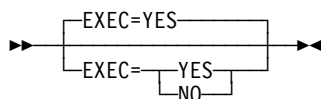
Default: YES

Specifies how the block handler is to be activated. If you code EXEC=YES, the block handler is activated during NCP initialization. If you code EXEC=NO, the block handler must be activated by a request from the host.

FEATURE

```
                  ┌─FEATURE=NOGPLKUP─┐
►►──────────────┼──────────────────┼──────►◄
                  └─FEATURE=──┬─NOGPLKUP─┬──┘
                             └─GPLKUP───┘
```

Default: NOGPLKUP

Specifies how NCP is to determine which device responds to a poll.

GPLKUP

Specifies that NCP is to search a table associated with this cluster to find the responding device. NCP searches only the cluster controller addresses that are in use, that is, those specified by ADDR on the TERMINAL definition statement. If you specify FEATURE=GPLKUP, you do not have to code a TERMINAL definition statement for each unused address on the cluster controller.

NOGPLKUP

Specifies that NCP is to use the polling response as an index value to locate the responding device. If you specify FEATURE=NOGPLKUP, code a separate TERMINAL definition statement for each address on the cluster controller even if no device is attached to that address. Code these TERMINAL definition statement in ascending address order.

GPOLL
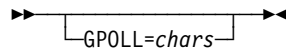
```
►►──────────────────────►◄
     └─GPOLL=chars─┘
```

Default: None

Specifies that NCP is to use the general polling procedure for this station and specifies the polling characters to be assigned to the control unit of the station. If you omit GPOLL, devices must be polled individually.

GPOLL is required if this CLUSTER definition statement represents an IBM 3271. For the ADDR keyword of each TERMINAL definition statement that defines a 2980, code the addressing characters assigned to that 2980. Because 2980s cannot be individually polled, the GPOLL keyword is not valid.

IBM 3270 Series (except 3275): If general polling is required, code the general-polling character POLL on CLUSTER. In addition, code a TERMINAL definition statement following this definition statement for each terminal address on the control unit. Code all addresses from the lowest address to the highest address, regardless of whether some intermediate addresses are unused. For example, if terminals are attached to the third, fourth, eighth, and ninth addresses, you code nine TERMINAL definition statements; the first definition statement represents the first (lowest) address and the last definition statement represents the ninth address.

Every attached terminal must be represented by a TERMINAL definition statement. Entering data from any terminal that is not represented cancels the general-polling operation for the control unit.

In the POLL and ADDR keywords on TERMINAL, code the polling and addressing character assigned to the device.
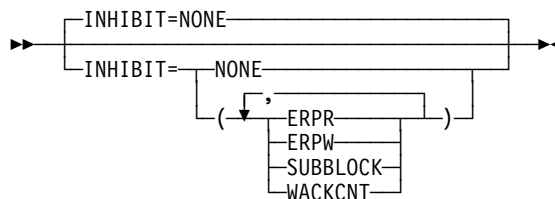
>   If an IBM 3277 is used as an input-only device and you specify general-polling characters on CLUSTER, omit the ADDR and POLL keywords from TERMINAL for that IBM 3277.

IBM 3275: If general polling is required, code the general-polling character in the GPOLL keyword of this definition statement. Also, directly following this CLUSTER definition statement, code a single TERMINAL definition statement and specify TERM=3275. If you want the IBM 3275 individually polled and addressed, specify the polling and addressing characters in the POLL and ADDR keywords on TERMINAL. If you specify addressing characters in the ADDR keyword, you must also code polling characters in the POLL keyword. (The reverse is not true.)

IBM 2972: Code the general-polling characters on the GPOLL keyword on CLUSTER. In addition, code a TERMINAL definition statement following this definition statement for each terminal address to the highest address, regardless of whether some intermediate addresses are unused. For example, if terminals are attached to the third, fourth, eighth, and ninth addresses, you code nine TERMINAL definition statements; the first definition statement represents the first (lowest) address and the last definition statement represents the ninth address.

INHIBIT

```
                  ┌─INHIBIT=NONE───────────────────────────┐
►►────────────────┤                                        ├────►◄
                  └─INHIBIT=─┬─NONE────────────────────┬────┘
                            │         ┌─,─────────┐    │
                            └─(───◄────┬─ERPR─────┬──)─┘
                                      ├─ERPW─────┤
                                      ├─SUBBLOCK─┤
                                      └─WACKCNT──┘
```

Default: NONE
Example: INHIBIT=(SUBBLOCK,ERPW)

Specifies which NCP facilities are to be inhibited when NCP starts running after being loaded into the controller. You can code the suboperands in any order. Separate the suboperands by commas.

WACKCNT
   Specifies that the wait-before-transmit positive acknowledgment (WACK) limit specified by the WACKCNT keyword on the GROUP definition statement is inhibited. The first WACK received from the station causes NCP to return the request to the host processor rather than respond to that and subsequent WACKs with an ENQ character.

SUBBLOCK
   Specifies that the subblocking indicated by TRANSFR on the LINE (or GROUP) definition statement is to be inhibited. If the number of buffers defined by TRANSFR are filled by received text, NCP terminates the receiving operation as though the cutoff limit (specified by the CUTOFF keyword) were reached. It is recommended that you specify INHIBIT=SUBBLOCK for BSC 3270 devices.

   Note: If you do not inhibit subblocking, code a value of 2 or more for TRANSFR.

ERPR
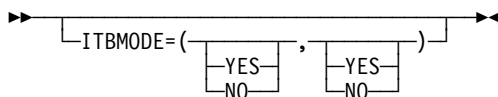   Specifies that recovery procedures for text-read errors are to be inhibited.

ERPW
Specifies that recovery procedures for text-write errors are to be inhibited.

NONE
Specifies that no NCP facilities are to be inhibited.

ITBMODE

```
►►──┬─────────────────────────────────┬──►◄
     └─ITBMODE=(──┬─────┬──,──┬─────┬──)─┘
                  ├─YES─┤     ├─YES─┤
                  └─NO──┘     └─NO──┘
```

Use: BSC devices in ITB mode
Default: See text

Specifies how NCP is to handle intermediate text block (ITB) characters in text received from the station or the access method.

If you code GPOLL on this CLUSTER definition statement, the default is ITBMODE=(NO,NO); otherwise, there is no default.

The first suboperand specifies whether NCP is to insert an error information block (EIB) character following each ITB character received from the station.

The second suboperand specifies whether an EIB character follows each ITB character received from the access method. If you specify YES, NCP removes the character following each ITB character before sending the data to the station.

LGRAPHS
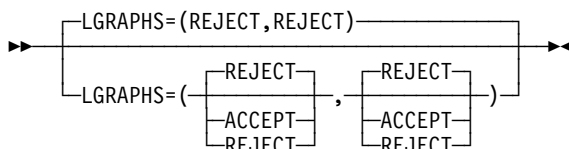
```
              ┌─LGRAPHS=(REJECT,REJECT)───────────┐
►►────────────┼───────────────────────────────────┼──►◄
              │          ┌─REJECT─┐     ┌─REJECT─┐  │
              └─LGRAPHS=(─┼────────┼──,──┼────────┼─)┘
                         ├─ACCEPT─┤     ├─ACCEPT─┤
                         └─REJECT─┘     └─REJECT─┘
```

Default: REJECT

Specifies whether NCP is to accept or reject leading graphics received from the station.

The first suboperand applies to read operations, and the second suboperand applies to write operations.

LMODADR

```
►►──┬──────────────────────────────────────┬──►◄
    └─LMODADR=(address1──────────────)─┘
                      └─,address2─┘
```
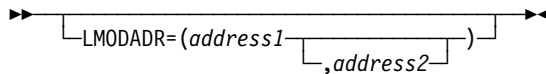
Default: See text

Specifies the local modem addresses used on the line going to the cluster controller. LMODADR is valid on the CLUSTER statement only if you specify LNCTL=BSC on the GROUP definition statement and LPDATS=LPDA2 on the LINE definition statement.

address1 and *address2* must be hexadecimal values ranging from X'0' to X'F'. *address1* specifies the local modem address on link segment 1; *address2* specifies the local modem address on link segment 2. *address2* is valid only if SEGMENT=2.

The default is LMODADR=01 if SEGMENT=1. The default is LMODADR=(01,02) if SEGMENT=2.

NEXT
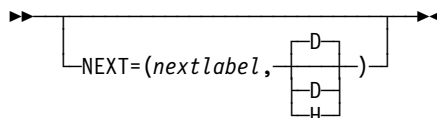
```
►►──┬─────────────────────────────┬──►◄
    │                    ┌─D─┐     │
    └─NEXT=(nextlabel,───┼─D─┼──)─┘
                         └─H─┘
```

Specifies a convention for constructing unique labels for the definition statements that will be copied when you use the AUTOCOPY keyword on the LINE definition statement. NEXT is valid only when you code AUTOCOPY on the previous LINE definition statement. NEXT is required when you code AUTOCOPY on the LINE definition statement.

See AUTOCOPY on the LINE definition statement on page 2-196 for more information. See the *NCP, SSP, and EP Resource Definition Guide* for more information about using NEXT.

nextlabel

   Default: None

   Specifies the label for the first duplicated copy of this CLUSTER definition statement. *nextlabel* is required when you code AUTOCOPY on the LINE definition statement. The label consists of 1 to 8 characters; the first character must be an alphabetical character from A to Z, #, or @. NDF uses *nextlabel* to determine what positions in the label to use for incrementing label names and the amount of the increment. The value you enter for *nextlabel* must be the same length as the statement label, and must differ from the label in such a way that NDF can identify a numeric field to use for incrementing. You can use only one field for incrementing. This field can increase in length as it is incremented, but it cannot increase so much that the total label length becomes greater than 8 characters.

The format of the statement *label* is

<p>lo_num_field<s>

The format of *nextlabel* is

<p>next_num_field<s>

where:

- *label* is the name specified on this CLUSTER definition statement.
- *nextlabel* is the label to be given to the first copy of this CLUSTER definition statement.
- *p* is the prefix of the numeric field and must be the same for both labels.
- *s* is the suffix of the numeric field and must be the same for both labels.

NDF subtracts lo_num_field from next_num_field to obtain the value it uses to increment the numeric field for the next label.

The numeric fields cannot differ in length, but must differ in value. next_num_field must be greater than lo_num_field.

If NDF cannot determine how to increment a label, it issues an error message.

<u>D</u>│H

Default: D

Specifies whether the increment and the field to increment are decimal (D) or hexadecimal (H). What you specify here does not need to match what you code for AUTOCOPY or for NEXT on other resource definition statements being replicated.

NPACOLL

```
                  ┌─NPACOLL=NO─────────────────────┐
►►────────────────┴─NPACOLL=─┬─(YES─────────────)─┬─►◄
                             │       └─,EXTended─┘ │
                             └─NO──────────────────┘
```

Default: NO

Specifies whether performance data for this cluster is eligible for collection by the network performance analyzer. NPACOLL=YES is valid only for the IBM 3271 and 3275 Cluster Controllers when NPACOLL=YES is coded on the LINE definition statement.

EXTENDED
Indicates that the form of the data collected by the network performance analyzer is different from the form used by NCP.

PT3EXEC

```
              ┌─PT3EXEC=NO──────┐
►►────────────┼─────────────────┼──►◄
              └─PT3EXEC=─┬─YES─┬─┘
                         └─NO──┘
```
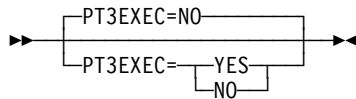
Use: Block handling routines
Default: NO

Specifies whether a block handler set that runs at point 3 is to be associated with this station. See the BHSET definition statement on page 2-4 for more information.

PT3EXEC is ignored if you omit the BHSET definition statement or if you code BHSET=NONE, BHEXEC=PT3, or BHEXEC=ALL on this CLUSTER definition statement.

SEGMENT

```
►►──┬───────────────────┬──►◄
    └─SEGMENT=─┬─1─┬─────┘
               └─2─┘
```
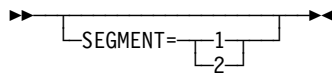
Default: See text

Specifies which link segment this resource is on, which allows NCP to set up LPDA2 tests accordingly. SEGMENT is valid only if you specify LPDATS=LPDA2 on the LINE definition statement.

SEGMENT=1 indicates that this resource is on segment number 1; SEGMENT=2 indicates that this resource is on segment number 2. SEGMENT=2 is valid only if you specify TAILING=YES on the LINE definition statement.

The default is SEGMENT=1 if TAILING=NO and SEGMENT=2 if TAILING=YES.

COMP Definition Statement

Usage: BSC or start-stop devices. One for each additional input or output compo-
nent of a device.

Position: Must follow TERMINAL definition statement or other COMP definition
statement.

Keywords: The keywords for this definition statement are equivalent to, and used
in the same way as, the corresponding keywords of the TERMINAL definition state-
ment. See the TERMINAL definition statement for keyword descriptions.

Specifying Lower-Level Keywords on Higher-Level Definition Statements:
Except for the POLL and ADDR keywords, you can define the keywords of the
COMP definition statement on the TERMINAL, LINE, or GROUP definition state-
ment. Table 1-3 on page 1-12 shows which keywords you can specify at a higher
level.

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the COMP definition statement
Default: None

Provides a name for the component. *symbol* can be any valid assembler language
symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to
avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process
creates. If AUTOLINE is specified, you must embed the first suboperand of the
ADDRESS keyword on the LINE definition statement in the symbol.

DATETIME Definition Statement

Usage: BSC or start-stop devices. One allowed for each block handler.

Position: Must follow EDIT definition statement if EDIT is coded.

Keywords: Table 2-12 shows the keywords described under this definition statement.

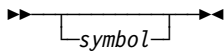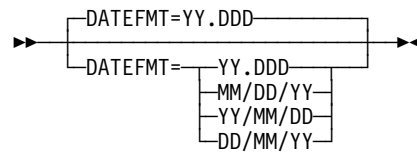*Table 2-12. Keywords on the DATETIME Definition Statement*

Keyword	Page	Required	Sift	Default	Tuning*
DATE	2-76			YES	
DATEFMT	2-77			YY.DDD	
INSERT	2-77			ALL	
PT2EXEC	2-77			BEFORE	
TIME	2-78			YES	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►──┬────────┬──►◄
    └─symbol─┘
```

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

DATE

```
        ┌─DATE=YES─┐
►►──────┼──────────┼───────►◄
        └─DATE=─┬─YES─┬─┘
                └─NO──┘
```
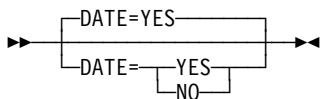
Default: YES

Specifies whether NCP is to insert the current date in a block of data.

Do not code both DATE=NO and TIME=NO.

DATEFMT

```
                 ┌─DATEFMT=YY.DDD─┐
  ▶▶─────────────┴────────────────┴──────────▶◀
           └─DATEFMT=──┬─YY.DDD─┬──┘
                       ├─MM/DD/YY─┤
                       ├─YY/MM/DD─┤
                       └─DD/MM/YY─┘
```

Default: YY.DDD

Specifies the format for the date inserted by the DATE keyword.

Code DATEFMT=YY.DDD (or omit the keyword) if you want the date to appear in Julian format (the year followed by the day of the year). For example, March 31, 1995, the 90th day of 1995, would appear as 95.090.

Code DATEFMT=MM/DD/YY if you want the date to appear in the month/day/year format. For example, April 1, 1995 would appear as 04/01/95.

Code DATEFMT=YY/MM/DD if you want the date to appear in the year/month/day format. For example, April 1, 1995 would appear as 95/04/01.

Code DATEFMT=DD/MM/YY if you want the date to appear in the day/month/year format. For example, April 1, 1995 would appear as 01/04/95.

INSERT

```
          ┌─INSERT=ALL─┐
  ▶▶──────┴────────────┴──────▶◀
     └─INSERT=──┬─ALL───┬─┘
               └─FIRST─┘
```
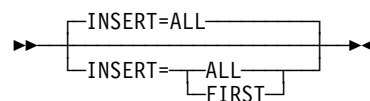
Default: ALL

Specifies whether NCP is to insert the date and time in the first block of each message or in all blocks.

PT2EXEC

```
          ┌─PT2EXEC=BEFORE─┐
  ▶▶──────┴────────────────┴──────▶◀
     └─PT2EXEC=──┬─BEFORE─┬─┘
                └─AFTER──┘
```
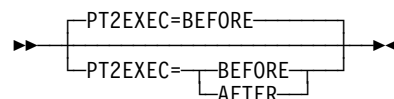
Default: BEFORE

Specifies, for a routine run at point 2, whether the routine inserts the date and time before the I/O operation (PT2EXEC=BEFORE) or after the I/O operation (PT2EXEC=AFTER).

TIME

```
                           ┌─TIME=YES─────────┐
 ►►────────────────────────┼──────────────────┼───►◄
                           └─TIME=──┬─YES─┬────┘
                                    └─NO──┘
```
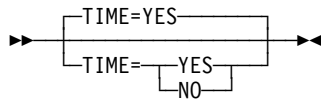
Default: YES

Specifies whether NCP is to insert the current time in a block of data. The time is always in the format *hh.mm.ss*, using the continental (24-hour) form. For example, 07.42.18 represents 7:42:18 a.m. and 19.42.18 represents 7:42:18 p.m.

Do not code both DATE=NO and TIME=NO.

DELETE Definition Statement

Usage: For dynamic reconfiguration. One for each physical unit or logical unit you want to delete from an existing link or physical unit, respectively.

Position: In a separate generation definition. If you specify a link, the DELETE definition statement must precede a PU definition statement. If you specify a physical unit, DELETE must precede an LU definition statement.

Keywords: Table 2-13 shows the keyword described under this definition statement.

Table 2-13. Keywords on the DELETE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
FROM	2-79	Required			

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►─┬────────┬─►◄
   └─*symbol*─┘

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

FROM

►►──FROM=*name*──►◄

Use: Required when you code the DELETE definition statement
Default: None

If you want to delete a physical unit dynamically, FROM specifies the name of the LINE definition statement from which the physical unit is to be deleted.

If you want to delete a logical unit dynamically, FROM specifies the name of the physical unit from which the logical unit is to be deleted.

For an example of how to code the DELETE definition statement for dynamic reconfiguration, see the *NCP, SSP, and EP Resource Definition Guide.*

DIALSET Definition Statement

Usage: BSC or start-stop devices on switched data links. One for each dial set defined.

Position: Must precede first GROUP definition statement.

Keywords: Table 2-14 shows the keywords described under this definition statement.

Table 2-14. Keywords on the DIALSET Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
DIALALT	2-80				
LINES	2-81	Required			
QLIMIT	2-81			1	
QLOAD	2-82			0	
RESERVE	2-82			0	

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the DIALSET definition statement
Default: None

Provides a name for the dial set. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

DIALALT

►►─┬──────────────┬─►◄
　　└─DIALALT=*name*─┘

Default: None

Specifies an alternate dial set for the dial set defined by this DIALSET definition statement. *name* is the name of the DIALSET definition statement for the alternate dial set. The alternate dial set must consist of the same type of lines as the primary dial set.

The DIALSET definition statement specified by the DIALALT keyword must immediately follow the DIALSET definition statements that define this dial set. Omit DIALALT from the last of a chain of DIALSET definition statements. Do not use DIALALT to specify the name of the first DIALSET definition statement in the chain.

LINES

```
►►──LINES=(──┬─────┬─name──┬──)──►◄
             └──,──┘
```

Use: Required when you code the DIALSET definition statement
Default: None

Specifies the switched lines that compose the dial set. *name* is the name of the LINE definition statement for an included line. Only a line whose LINE definition statement specifies CALL=OUT or CALL=INOUT can be included in a dial set. No line can appear in more than one dial set.

You can specify a maximum of 255 characters for LINES, including the beginning and ending parentheses and all commas. This limit applies regardless of how many line names you code within the keyword. If you need more than 255 characters to complete the list, code one or more additional DIALSET definition statements (omitting *symbol*) after the first. Code the remaining line names in LINES. No other keywords can be specified in the additional DIALSET definition statements.

QLIMIT

```
         ┌─QLIMIT=1────────┐
►►───────┼─────────────────┼───►◄
         └─QLIMIT=requests─┘
```

Default: 1
Minimum: 0
Maximum: 255

Specifies the maximum number of callout requests NCP is to allow to accumulate in the queue for this dial set. When this limit is reached, NCP returns any further callout requests it receives to the host processor. NCP continues to reject callout requests until the number of requests in the queue drops below the value of QLIMIT.

If you code QLIMIT=0, NCP rejects all callout requests when no line is available.
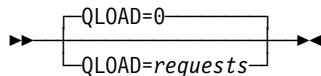
QLOAD

```
              ┌─QLOAD=0──────────┐
►►────────────┴──────────────────┴──►◄
              └─QLOAD=requests───┘
```

Default: 0
Minimum: 0
Maximum: 255

Specifies the maximum number of callout requests NCP is to allow to accumulate
in the queue for this dial set before using a line from the alternate dial set. If you
code QLOAD=0 (or omit QLOAD) and define an alternate dial set, NCP uses a line
from the alternate dial set if no line in the primary dial set is available.

The value of QLOAD must be less than the value of QLIMIT. Otherwise, the queue
of unfulfilled callout requests cannot use the alternate dial set.

RESERVE
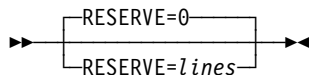
```
              ┌─RESERVE=0──────┐
►►────────────┴────────────────┴──►◄
              └─RESERVE=lines──┘
```

Default: 0
Minimum: 0
Maximum: 255

Specifies how many lines in the dial set are to be reserved for incoming calls from
devices. If you code RESERVE=0 (or omit RESERVE), no lines are reserved.
When all lines are busy with outgoing calls, no devices can call the communication
controller.

| # DLCI Definition Statement

| **Usage:**   Frame-relay physical lines. Specifies the DLCI(s) for which the
| statement's keyword parameters apply.

| A DLCI statement may be coded for each DLCI that can be supported by the phys-
| ical line. However, a DLCI statement is not required for every DLCI supported by
| the physical line.

| No more than one DLCI statement may be coded for a given DLCI on a physical
| line.

| Only the ID keyword is required on a DLCI statement. All other keywords are
| optional.

| **Position:**   Must follow a frame-relay physical LINE statement.

| **Keywords:**   Table 2-15 shows the keywords described under this definition state-
| ment.

| *Table 2-15. Keywords on the DLCI Definition Statement*

Keyword	Page	Required	Sift	Default	Tuning*
ID	2-84	Required			
CIR	2-85			NO	Yes*
BALANCE	2-86				Yes*

| ***Tuning:**

| Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
| *NTuneMON V2R5 User's Guide* for more information.

| # Keyword Descriptions

| # symbol

| ►►──*symbol*──►◄

| Default: None

| Provides a name for the DLCI statement. *symbol* can be any valid assembler lan-
| guage symbol that does not duplicate any existing symbol.

| See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to
| avoid when naming resources.

| The symbol appears in the label cross-reference listing that the generation process
| creates.

| **ID**

```
                        ┌─────────D─────────┐
    ►►─┬──────────────────────────────────────────────┬─►◄
       │                      ┌─D──┐                   │
       └─ID=─┬─(dlci,─┬─type─┬──)──┤                   │
             │        ├──D──┤                          │
             │        ├──H──┤                          │
             │        └─DD──┘                          │
             ├─dlci₁─dlci₂───────┘                     │
             └─DEFAULT───────────┘
```
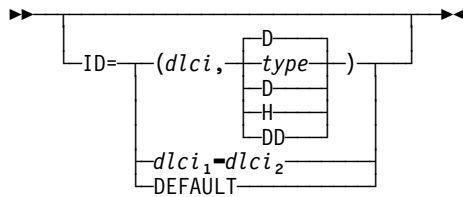
| Default: None
| Minimum: (16,D) or (10,H)
| Maximum: The MAXDLCI value for this frame-relay interface. See MAXDLCI on
| the LINE definition statement on page 2-245.

| Specifies the DLCI to which the DLCI statement keywords apply. ID can be speci-
| fied in either of three number bases:

| **D** Decimal

| **H** Hexadecimal

| **DD** Dotted decimal

| ID can specify an individual DLCI or can specify a sequential DLCI range by using
| **-** as the range indicator. Only decimal values can be used to specify a DLCI
| range.

| If a single DLCI is defined with no type specified, decimal (D) is assumed.

| If a DLCI range is specified, then both $dlci_1$ and $dlci_2$ must be within the supported
| range of ID values. $dlci_2$ must not be less than $dlci_1$. Specifying a DLCI range
| allows you to define the parameter values for each DLCI included within the range.

| For certain keywords, coding ID=DEFAULT allows you to define parameter values
| to be used for all DLCIs for which you have not coded a specific DLCI statement.
| Refer to each keyword for additional information.

CIR

```
                        ┌─CIR=(1,1,.1)─────────────────┐
►►──────────────────────┴──────────────────────────────┴──►◄
        │        ┌─1─┐     ┌─1─┐      ┌─.1─┐
        └─CIR=(──┤   ├──,──┤   ├──,───┤    ├──)
                 └─Bc┘     └─Be┘      └─Tc─┘
```
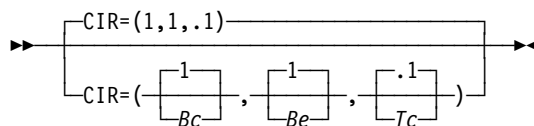
Use: 3746 Model 900 frame-relay DLCIs
Default: (1,1,.1)
Examples: DLCI ID=(17,D),CIR=(2,1,.1)
            DLCI ID=20-30,CIR=(2,1,.1)
            DLCI ID=DEFAULT,CIR=(1,1,.1)

Specifies the bandwidth management parameters to be used for a designated DLCI.

If ID=DEFAULT, then this CIR parameter specifies the default CIR parameter values for all DLCIs without a corresponding DLCI definition statement.

Note: CIR on the DLCI statement is valid only if CIR=YES on the LINE statement.

Bc

> Default: 1
> Minimum: 0
> Maximum: 64

> The value of the committed burst size (Bc) is multiplied by the value of the DATABLK keyword of the LINE statement to determine the maximum committed amount of data that a user can offer to the network during a time interval (Tc).

> *Bc* must be an integer.

> **Note:** ($Bc + Be$) > 0.

Be

> Default: 1
> Minimum: 0
> Maximum: 64

> The value of the excess burst size (Be) is multiplied by the value of the DATABLK keyword of the LINE statement to determine the maximum allowed amount of data by which a user can exceed the committed burst size (Bc) during a time interval (Tc).

> *Be* must be an integer.

> **Note:** ($Bc + Be$) > 0.

Tc

> Default: .1
> Minimum: .1
> Maximum: 25.5

> Specifies the committed rate measurement interval (Tc), which is defined as the time interval during which the user is allowed to send only the committed amount of data (Bc) and the excess amount of data (Be).

| Valid values for *Tc* are in increments of a tenth of a second.

| **BALANCE**
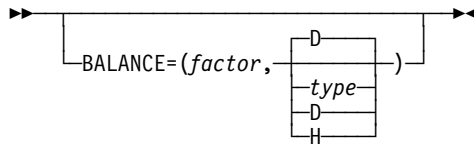
| Use: 3745 frame-relay DLCIs
| Default: None
| Minimum: (1,D)
| Maximum: (32,D) or (20,H)
| Examples: DLCI ID=(10,D),BALANCE=1
| DLCI ID=18-36,BALANCE=2

| Specifies the balancing factor to be applied to the DLCI(s) indicated by the ID
| keyword.

| Specify a balancing factor when you want that defined DLCI(s) to participate in bal-
| ancing of peripheral BAN connections (established through incoming call request)
| across all DLCIs configured with the same BAN DLCI-MAC address in the BAN
| router(s).

| ID=DEFAULT **cannot** be used to specify a default balancing factor for DLCIs.

| **Notes:**

| 1. A DLCI balancing factor can be specified only if:

| • The buffer size (BFRS on the BUILD statement) is 100 or greater.
| • The physical line can support peripheral BAN connections
| (FRELAY=(PHYSICAL,PERIPHERAL|ANY).

| 2. A balancing factor can be specified for a DLCI associated with a subarea FRTE
| or FHSP resource. However, connection balancing will not be performed for
| that DLCI.

EDIT Definition Statement

Usage: BSC or start-stop devices. One allowed for each block handler.

Position: Must appear before the DATETIME definition statement if DATETIME is coded.

Keywords: Table 2-16 shows the keyword described under this definition statement.

Table 2-16. Keywords on the EDIT Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
BKSP	2-87			16	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►──┬────────┬──►◄
    └─symbol─┘
```
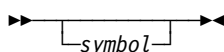
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

BKSP

```
      ┌─BKSP=16───┐
►►──┴─BKSP=char─┴──►◄
```
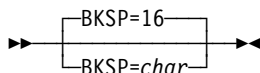
Default: 16

Defines the text-canceling (backspace) character. The text-canceling character allows the keyboard operator to signal NCP to correct text incorrectly entered. *char* is the hexadecimal representation of the text-canceling character.

If you omit BKSP, NCP uses X'16' (the hexadecimal representation of the EBCDIC backspace [BS] character) as the text-canceling character. When you code a value for BKSP, do not code single quotes ('') or framing characters of any type.

ENDAUTO Definition Statement

Usage: Required when you code the AUTOLINE or AUTOCOPY keyword on the LINE definition statement.

Position: ENDAUTO delimits the line group that is to be replicated.

Keywords: This definition statement has no keywords.

ENDBH Definition Statement

Usage: BSC or start-stop devices. One for each block handler.

Position: Must appear at the end of block handler.

Keywords: This definition statement has no keywords.
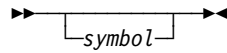
symbol

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

FRSESET Definition Statement

Usage: Frame-relay physical lines and NTRI frame handler logical lines. Defines a set of frame handler subports. This subport set defines the internal segment of a frame-relay permanent virtual circuit (PVC) between a frame-relay subport on one frame-relay physical line or NTRI frame handler logical line and a frame-relay subport on another corresponding line. This path, which is called an *internal PVC segment*, is used to forward frames received over one frame handler subport for transmission over another frame handler subport. PVC segments are identified by the subport names defined by the PU definition statements in a frame-relay physical line group or a NTRI frame handler logical line group.

Position: Must follow all PU definition statements for frame handler subports and must precede the GENEND definition statement.

Keywords: Table 2-17 shows the keyword described under this definition statement.

Table 2-17. Keywords on the FRSESET Definition Statement

Keyword	Page	Required	Sift	Default	Tuning
SUBPORTS	2-91	Required			

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the FRSESET definition statement
Default: None

Provides a name for the frame-relay subport set. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

SUBPORTS

```
►►──SUBPORTS=(primary1,primary2,──────────────,──────────────)──►◄
                                 └─substitute1─┘   └─substitute2─┘
```

Use: Required when you code the FRSESET definition statement
Default: None
Example: SUBPORTS=(C04P28B,C04P16B,,C04P20B)

Defines a set of primary frame handler subports and optional substitute frame
handler subports. This subport set constitutes an internal PVC segment. An NCP
or 3746 Model 900 uses *substitute1* when *primary1* is not available and uses
substitute2 when *primary2* is not available.

Each subport you specify can have any combination of frame handler subports on a
separate frame-relay physical line or NTRI frame handler logical line. All frame
handler subports specified by SUBPORTS must be on different LINE definition
statements. All the subports in the set must have the same value for MAXFRAME
on the LINE definition statement.

Do not specify the same subport more than once in a subport set or on more than
one FRSESET definition statement.

You can define primary and substitute subports only on a line whose link-station
subport supports the local management interface (LMI) protocol. See LMI on the
PU definition statement on page 2-388 for more information.

3746 Model 900 Frame Relay: With NCP V7R2 and NCP V7R3, the subports in a
        subport set must all be 3746 Model 900 FHSPs or must all be 3745 FHSPs
        because internal frame switching between a 3746 Model 900 and NCP
        running on a 3745 is not supported. Starting with NCP V7R4, you can
        define an internal PVC segment between a 3745 and a 3746 Model 900 by
        specifying a set of frame handler subports, one on a 3745 frame-relay phys-
        ical line or NTRI frame handler logical line and another on a 3746 Model
        900 frame-relay physical line.

GENEND Definition Statement

| ***Usage:*** Required for NCP and EP.

Position: Must follow all other NCP definition statements unless you code the OPTIONS definition statement last.

Keywords: Table 2-18 shows the keywords described under this definition statement.

| ***Note:*** For EPSA, no keywords can be specified on the GENEND definition state-
| ment.

Table 2-18. Keywords on the GENEND Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
INCHI	2-96				
INCINIT	2-97				
INCLO	2-96				
INCL2HI	2-97				
INCL2LO	2-97				
INIT	2-94				
KEY0INC	2-98				
KEY0ORD	2-98				
LINOPT	2-98				
ORDHI	2-98				
ORDINIT	2-99				
ORDLO	2-98				
ORDL2HI	2-99				
ORDL2LO	2-99				
SRCHI	2-93				
SRCLO	2-93				
TMRTICK	2-95				
UACCTNG	2-95				
UGLOBAL	2-95				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

The keywords of this definition statement are described in the following groups.

Keywords used to include user-written code:

SRCHI   SRCLO

Keywords used to specify the entry point for user-written or NTRI modules:

INIT   TMRTICK   UACCTNG   UGLOBAL

Linkage editor keywords used to include user-written code:

INCHI	INCL2HI	KEY0ORD	ORDLO	ORDL2HI
INCLO	INCL2LO	LINOPT	ORDINIT	ORDL2LO
INCINIT	KEY0INC	ORDHI		

Note: User-defined control blocks, tables, and user-written code included using SRCHI and INCHI may reside above the 4MB boundary in controller storage. Use the SRCLO and INCLO keywords to specify information that must reside below the 4MB boundary (for example, executable code). Use the INCL2LO keyword to specify information that must reside below the 64KB boundary.

Keyword Descriptions

symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates. If you specify more than one GENEND definition statement, code *symbol* only on the first GENEND definition statement.

Keywords Used to Include User-Written Code

The following keywords provide an NCP generation procedure with the library member names of user-written modules. You can specify as many symbols as necessary for each keyword. However, the NDF has a limit of 255 characters per keyword, which includes commas and framing parentheses.

SRCHI and SRCLO

These keywords specify the source code for user-defined control blocks and tables to be assembled with NCP control blocks and tables. The source code must be in macro or open code form using communication controller assembler instructions.

The values you code for SRCHI and SRCLO represent the member names of the macros as they are cataloged in the user macro library or NCP phase 2 macro library (SNCPMAC1).

SRCHI defines source code that can reside anywhere in communication controller storage. Do not use SRCHI for executable code.

If you specify user-written line control (the LEVEL2 keyword on the GROUP definition statement), do one of the following:

- Specify SRCLO

- Have a generation application load module pass link-edit data to the NDF link-edit cluster for ordering user control blocks

User-defined control blocks and tables included using the SRCHI keyword have been moved toward the end of the NCP load module and may be above the 4MB boundary. If you want information to reside below the 4MB boundary, include that information with the SRCLO keyword. See the *NCP and SSP Customization Guide* for more information.

Keywords Used to Specify Entry Points for User-Written Code

The following keywords provide the NCP generation procedure with the entry points of user-written, NTRI, or frame-relay modules.

INIT
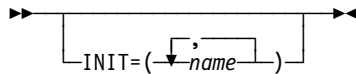
Specifies the entry points of user-written, NTRI, or 3745 frame-relay initialization routines. These routines must be object modules cataloged in the MVS object library or cataloged as type OBJ in the VSE library or members of the VM CMS text library.

If you use INIT to specify entry points for user-written modules, do one of the following:

- Code at least one of the following keywords:

    INCINIT
    INCHI
    INCLO

- Have a generation application pass data to the NDF link-edit clusters for INCINIT, INCHI, or INCLO using the NDF standard attachment facility

Table 2-19 shows the effect on INIT when you code certain keywords on the GROUP definition statement.

Table 2-19. NDF Values for INIT on the GENEND Definition Statement

When You Code This on GROUP	The Effect on INIT Is
ECLTYPE (NTRI only)	NDF adds INIT=ECLINIT.
FRELAY (3745 only)	NDF adds INIT=ECLINIT.

TMRTICK

```
►►──┬─────────────────────────┬──►◄
    │          ┌─,─┐          │
    └─TMRTICK=(─▼─name─┴─)────┘
```
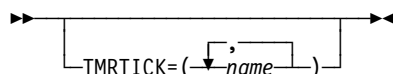
Specifies the entry point of up to five user-written, NTRI, or 3745 frame-relay timer-tick service routines. These routines must be object modules cataloged in the MVS object library or cataloged as type OBJ in the VSE library or members of the VM CMS text library.

To include the timer-tick routine in NCP, also specify either the INCL2HI or the INCL2LO keyword.

NTRI and 3745 Frame Relay: You do not need to specify INCL2HI or INCL2LO to include the timer-tick routine for NTRI or 3745 frame relay.

Table 2-20 shows the effect on TMRTICK when you code certain keywords on the GROUP definition statement.

Table 2-20. NDF Values for TMRTICK on the GENEND Definition Statement

When You Code This on GROUP	The Effect on TMRTICK Is
ECLTYPE (NTRI only)	NDF adds TMRTICK=ECLTICK.
FRELAY (3745 only)	NDF adds TMRTICK=ECLTICK.

UACCTNG

```
►►──┬─────────────────────────┬──►◄
    │          ┌─,─┐          │
    └─UACCTNG=(─▼─name─┴─)────┘
```
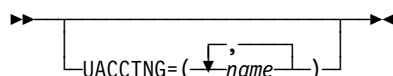
Specifies the entry points of user-written code or IBM special products accounting notification routines. These routines must be object modules cataloged in the MVS object library or cataloged as type OBJ in the VSE library or members of the VM CMS text library.

If you use UACCTNG to specify entry points for user-written modules, include the object modules in the generation definition in one of these ways:

* Use INCL2HI or ORDL2HI on the GENEND definition statement.
* Use USERGEN on the OPTIONS definition statement.

UACCTNG is valid only if you specify SESSACC=YES on the BUILD definition statement.

UGLOBAL
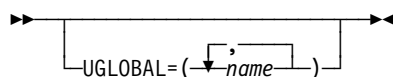
```
►►──┬─────────────────────────┬──►◄
    │          ┌─,─┐          │
    └─UGLOBAL=(─▼─name─┴─)────┘
```

Specifies the entry points of IBM special products or user-written code, NTRI, or 3745 frame-relay routines that act as global routers.

Table 2-21 shows the effect on UGLOBAL when you code certain keywords on the GROUP definition statement.

Table 2-21. NDF Values for UGLOBAL on the GENEND Definition Statement

When You Code This on GROUP	The Effect on UGLOBAL Is
ECLTYPE (NTRI only)	NDF adds UGLOBAL=ECLUGBL.
FRELAY (3745 only)	NDF adds UGLOBAL=ECLUGBL.

Linkage Editor Keywords Used to Include User-Written Code

The following keywords specify the names of macro library members that contain the linkage editor INCLUDE and ORDER statements for user-written object modules.

On an MVS or VM system, the object modules must be members of the user object library specified on the INCLUDE DD card or members of the NCP object library (SNCPMOD1). Every control section (CSECT) in the user object modules must be specified on a linkage editor ORDER statement. Any CSECT not specified on an ORDER statement will not be in the correct position in the load module and will be overlaid during initialization of NCP.

In a VSE system, the object modules must be cataloged as type OBJ in a VSE sublibrary. Specify only the INCLUDE keywords of this definition statement for a VSE system. ORDER keywords are not valid.

INCHI and INCLO
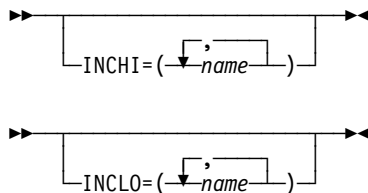
```
►►──┬──────────────────────┬──►◄
    │          ┌─;──┐       │
    └─INCHI=(──▼─name─┴──)──┘
```

```
►►──┬──────────────────────┬──►◄
    │          ┌─;──┐       │
    └─INCLO=(──▼─name─┴──)──┘
```

These keywords specify the macro library members (type OBJ for VSE) that contain linkage editor INCLUDE statements for object modules that are not level 2 or level 3 code. If you specify VIRTUAL=YES on a GROUP definition statement or include an NCPNAU definition statement in the generation, do one of the following:

- Specify at least one of these keywords (INCHI or INCLO)

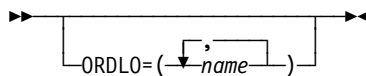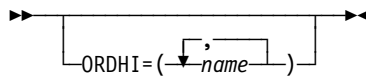- Have a generation application load module which passes data to the NDF link-edit clusters for INCHI or INCLO

INCHI names the object modules that can reside anywhere in the controller storage. These modules run with a storage protect key of 1.

INCLO names those object modules that must reside in the lower 64KB of controller storage. These modules run with a storage protect key of 0.

If you include user block handling routines (UBHR definition statements) in the generation and specify INCHI, the generation process assumes that the block handling routine linkage editor control statements are included in the members specified on INCHI.

User-written code included in INCHI may have been moved toward the end of the NCP load module and may be above the 4MB boundary. NCP's executable code must remain below the 4MB boundary because the LA, BAL, and BLG instructions can only address 22 bits or 4MB. You can specify information that needs to reside below the 4MB boundary (for example, executable code) with the INCLO keyword. Use INCL2LO if you need to place user-written code near the beginning of the load module.

Modules placed with INCLO will be placed in the location previously used for INCHI.

Migration Aid Function: If the input generation definition is for a release before NCP V5R4, the migration aid function replaces the value of INCHI with question marks and issues a message asking if you still want to use INCHI.

INCINIT

Specifies the macro library members (type OBJ for VSE) that contain linkage editor INCLUDE statements for user-written initialization code. The modules included in NCP generation by this keyword are overlaid after the initialization is complete. If you do not want the user initialization code overlaid, specify the library members containing the INCLUDE statements in INCHI or INCLO.

INCL2HI and INCL2LO

These keywords specify the macro library members (type OBJ for VSE) that contain linkage editor INCLUDE statements for object modules that are level 2 code, level 3 code, or the user RECMS/NMVT module that NCP calls in level 5. These modules run with a storage protect key of 0.

INCL2HI names the object modules that can reside anywhere in communication controller storage.

INCL2LO names those object modules that must reside in the low 64KB of controller storage. If you specify user-written line control (LEVEL2 on a GROUP definition statement), do one of the following:

- Specify INCL2LO

- Have a generation application load module pass data to the NDF link-edit cluster for INCL2LO
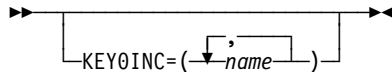
KEY0INC

```
►►─┬─────────────────────────┬─►◄
   │        ┌─,────┐          │
   └─KEY0INC=(─▼─name─┴─)─┘
```

Specifies the macro library members (type OBJ for VSE) that contain linkage editor
INCLUDE statements for object modules to be included in read-only storage.

KEY0ORD
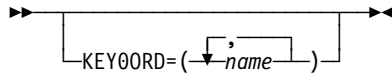
```
►►─┬─────────────────────────┬─►◄
   │        ┌─,────┐          │
   └─KEY0ORD=(─▼─name─┴─)─┘
```

MVS or VM: Specifies the macro library members that contain linkage editor
 ORDER statements for object modules to be included in read-only storage.

LINOPT

```
►►─┬──────────────┬─►◄
   └─LINEOPT=BSC─┘
```
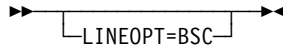
Specifies that binary synchronous line control modules are to be included in the
NCP load module. Use LINOPT if you have user-written code that requires BSC
line control, but have no real BSC devices defined. If you do not need BSC
devices, omit this keyword.

ORDHI and ORDLO

```
►►─┬───────────────────────┬─►◄
   │      ┌─,────┐          │
   └─ORDHI=(─▼─name─┴─)─┘
```

```
►►─┬───────────────────────┬─►◄
   │      ┌─,────┐          │
   └─ORDLO=(─▼─name─┴─)─┘
```

MVS or VM: These keywords specify the macro library members that contain
 linkage editor ORDER statements for object modules that are not level 2 or
 level 3 code.

ORDHI names the object modules that can reside anywhere in the commu-
nication controller storage. ORDHI is required if you specify INCHI on this
definition statement and TYPSYS=OS or TYPSYS=VM on BUILD.

ORDLO names those object modules that must reside in the lower 64KB of
controller storage. ORDLO is required if you specify INCLO on this defi-
nition statement and TYPSYS=OS or TYPSYS=VM on BUILD.

If user block handling routines (UBHR definition statement) are included in
the generation and you specify INCHI, the generation process assumes that
the linkage editor control statements of the block handling routine are
included in the members specified in INCHI and ORDHI.

Some NCP control blocks have been moved higher in the load module and
may be above the 4MB boundary. Executable code must remain below the

4MB boundary because the LA, BAL, and BLG instructions can only address 22 bits, or 4MB. You can specify information that needs to reside below the 4MB boundary (for example, executable code) with the ORDLO keyword.

Migration Aid Function: If the input generation definition is for a release before NCP V5R4, the migration aid function replaces the value of ORDHI with question marks and issues a message asking if you still want to use ORDHI.

ORDINIT
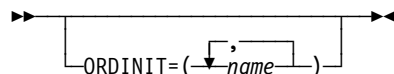
MVS or VM: Specifies the macro library members that contain linkage editor ORDER statements for user-written initialization code. ORDINIT is required and valid only if you code INCINIT on the GENEND definition statement and TYPSYS=OS or TYPSYS=VM on the BUILD definition statement.

The members specified in ORDINIT are overlaid after the initialization is complete. If you do not want the user initialization code to be overlaid, use the ORDHI and ORDLO keywords to specify the library members containing ORDER statements.

ORDL2HI and ORDL2LO
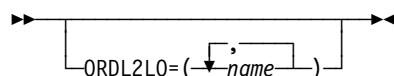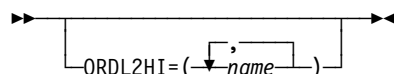
MVS or VM: These keywords specify the macro library members that contain linkage editor ORDER statements for object modules that are level 2 or level 3 code.

ORDL2HI names the object modules that can reside anywhere in communication controller storage. It is required if you specify INCH2HI and TYPSYS=OS or TYPSYS=VM on the BUILD definition statement.

ORDL2LO names the object modules that must reside in the lower 64KB of controller storage. It is required if you specify INCL2LO and TYPSYS=OS or TYPSYS=VM on the BUILD definition statement.

GROUP Definition Statement

Usage: Required. One for each physical line group, even if the group has only one telecommunication link.

Position: Must appear at beginning of each line group definition.

The order of the GROUP definition statements depends on the type of line control defined. BSC and start-stop line groups must appear before all other line groups in the generation definition. SDLC line groups must follow all BSC and start-stop line groups. Channel link groups must appear after all SDLC line groups. User line groups may precede or follow channel link groups.

Keywords: Table 2-22 shows the keywords described under this definition statement.

Table 2-22 (Page 1 of 2). Keywords on the GROUP Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ACTIVTO	2-105				Yes*
ADAPTER	2-106				
AUTOGEN	2-107				
BALANCE	2-200	5.D			
CAEXIT	2-107				
CHANLNK	2-108				
CHAREC	2-108			XONOFF	
COMPACB	2-109			NO	
COMPOWN	2-110			NO	
COMPSWP	2-111			NO	
COMPTAD	2-112			NO	
CRETRY	2-112				
DELAY	2-113			NO	
DIAL	2-113			NO	
DSRTO	2-114			51	
ECLTYPE	2-115				
EOB	2-116				
EOT	2-116				
ETHERNET	2-117				
FRAMING	2-117			11	
FRELAY	2-118				
ISDN	2-119				
KBDLOCK	2-120			80	
LEVEL2	2-120				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.
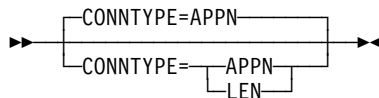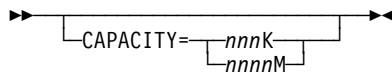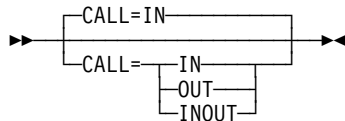
Table 2-22 (Page 2 of 2). Keywords on the GROUP Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
LEVEL3	2-121				
LEVEL5	2-122			USER	
LINEADD	2-123				
LINEAUT	2-123				
LNCTL	2-124				
LNKOWNER	2-125				
LSPRI	2-125			PU	
MODE	2-127				
NCST	2-127				
NETID	2-127				
NPARSC	2-128			NO	
PADCNT	2-128				
PECHAR	2-129			FF	
PHYPORT	2-129				
PHYRSRC	2-130				
PRODID	2-131				
PUTYPE	2-131				
QUIETCT	2-132			0	
REPLYTO	2-132				Yes*
RETRYTO	2-133				Yes*
RNRLIMT	2-134			3	Yes*
TEXTTO	2-134				Yes*
TIMER	2-135				
TTDCNT	2-136			15	
TYPE	2-136				
USERID	2-137				
VIROWNER	2-138				
VIRTUAL	2-139			NO	
WACKCNT	2-139			15	
WAKDLAY	2-140			2.2	
WTTYEOB	2-140				
WTTYEOT	2-140				
XIO	2-141				
XMITDLY	2-142				Yes*
X21NTWK	2-142			NO	Yes*
X21SW	2-143			NO	

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
NTuneMON V2R5 User's Guide for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements:
You can define most keywords of the COMP, CLUSTER, LINE, LU, PU, and TER-
MINAL definition statements on the GROUP definition statement. Table 1-3 on
page 1-12 shows which lower-level keywords you can define on GROUP.

If you code NPARSC=YES, NDF ignores all keywords defined on GROUP except
LNCTL, VIRTUAL, and CHANLNK.

VTAM Keywords: The following diagrams show the VTAM keywords that can be
coded on this definition statement. These keywords provide information to the
VTAM initialization process. They are not required in the generation definition you
use to generate NCP. See the *VTAM Resource Definition Reference* for
descriptions of these keywords and the *VTAM Network Implementation Guide* for
information on the VTAM initialization process.

VTAM keywords not listed below can be specified by coding VTM. as a prefix in
front of the VTAM keyword.

```
          ┌─ANSWER=ON─────────┐
►►────────┤                   ├──►◄
          └─ANSWER=──┬─ON──┬──┘
                     └─OFF─┘
```

```
►►────────────────────────────────────────────────►◄
          └─ASLENT=associated_LU_table_entry_name─┘
```

```
►►──────────────────────────────────────►◄
          └─ASLTAB=associated_LU_table_name─┘
```

```
          ┌─CALL=IN─────────────┐
►►────────┤                     ├──►◄
          └─CALL=──┬─IN────┬────┘
                   ├─OUT───┤
                   └─INOUT─┘
```

```
►►──────────────────────────►◄
          └─CAPACITY=──┬─nnnK──┬─┘
                       └─nnnnM─┘
```

```
          ┌─CONNTYPE=APPN───────┐
►►────────┤                     ├──►◄
          └─CONNTYPE=──┬─APPN─┬─┘
                       └─LEN──┘
```

```
►►──────────────────►◄
          └─COSTBYTE=n─┘
```

```
►►──────────────────►◄
          └─COSTTIME=n─┘
```

```
►►────────────────────►◄
          └─CPCP=──┬─YES─┬─┘
                   └─NO──┘
```

```
      ┌─DISCNT=NO────────────────────────────────────┐
►►─────┤                                              ├─►◄
       │           ┌──────────────┐                   │
       └─DISCNT=─┬─(─YES─┬─,─F────┬──────)─┐           │
                 │       │   ┌─F──┐        │           │
                 │       └─,─┤    ├────────┘           │
                 │           └─NF─┘                    │
                 └─NO────────────────────────────────┘
```

```
►►─┬──────────────────────────────────────────────────┬─►◄
   └─DLOGMOD=default_logon_mode_entry_name─────────────┘
```

```
►►─┬─────────────────────┬─►◄
   └─DYNADJCP=─┬─YES─┬────┘
               └─NO──┘
```

```
►►─┬──────────────────┬─►◄
   └─DYNLU=─┬─YES─┬────┘
            └─NO──┘
```

```
   ┌─DYNPU=NO──────────┐
►►─┤                   ├─►◄
   └─DYNPU=─┬─NO──┬─────┘
            └─YES─┘
```

```
   ┌─DYNPUPFX=CN───────────┐
►►─┤                       ├─►◄
   └─DYNPUPFX=string───────┘
```

```
►►─┬──────────────────────────┬─►◄
   │          ┌─,─────┐        │
   └─FEATUR2=(─▼─keyword─┬─)────┘
```

```
►►─┬───────────────┬─►◄
   └─HPR=─┬─YES─┬───┘
          └─NO──┘
```

```
►►─┬───────────────────┬─►◄
   └─INCLUD0E=─┬─NO──┬──┘
               └─YES─┘
```

```
   ┌─ISTATUS=ACTIVE──────────┐
►►─┤                         ├─►◄
   └─ISTATUS=─┬─ACTIVE───┬────┘
              └─INACTIVE─┘
```

```
►►─┬──────────────────────────────────────────┬─►◄
   └─LOGAPPL=application_program_name──────────┘
```

```
►►─┬────────────────────────────────┬─►◄
   └─LOGTAB=interpret_table_name─────┘
```

```
►►─┬──────────────────────────────────────────┬─►◄
   └─MDLENT=model_name_table_entry_name────────┘
```

```
►►─┬──────────────────────────────────┬─►◄
   └─MDLTAB=model_name_table_name──────┘
```

```
►►─┬──────────────────────────────────┬─►◄
   └─MODETAB=logon_mode_table_name─────┘
```

GROUP

```
>>──┬─────────────────────────┬──><
    └─NATIVE=──┬─NO──┬─────────┘
              └─YES─┘


>>──┬──────────────────┬──><
    └─NN=──┬─YES─┬──────┘
          └─NO──┘


>>──┬──────────────────────────────────┬──><
    │              ┌──,──────────┐      │
    └─OWNER=(──────▼─owner_name──┴──)───┘


      ┌─PDELAY=1───┐
>>────┼────────────┼──><
      └─PDELAY=n───┘


      ┌─PU=YES───────┐
>>────┼──────────────┼──><
      └─PU=──┬─YES─┬──┘
            └─NO──┘


>>──┬──────────────────────────┬──><
    └─REGISTER=──┬─NO───────┬───┘
                ├─NETSRVR─┤
                └─DIRSRVR─┘


>>──┬────────────────────────────┬──><
    └─SECURITY=──┬─UNSECURE─┬─────┘
                ├─PUBLIC───┤
                ├─UNDERGRO─┤
                └─SECURE───┘


      ┌─SHOLD=NO──────┐
>>────┼───────────────┼──><
      └─SHOLD=──┬─NO──┬┘
              └─YES─┘


>>──┬───────────────────┬──><
    └─SPAN=span_name─────┘


      ┌─SSCPFM=FSS───────────┐
>>────┼──────────────────────┼──><
      └─SSCPFM=──┬─FSS────┬───┘
               ├─USSSCS─┤
               ├─USS3270┤
               ├─USS3275┤
               ├─USS3780┤
               └─USSNTO─┘


>>──┬──────────────────┬──><
    └─TGP=tgp_name──────┘


      ┌─TOPO=CONNECT─────┐
>>────┼──────────────────┼──><
      └─TOPO=──┬─CONNECT─┬┘
             └─NONE────┘


>>──┬───────────┬──><
    └─UPARM1=n──┘
```

```
►►────────────────────►◄
      └─UPARM2=n─┘
```

```
►►────────────────────►◄
      └─UPARM3=n─┘
```

```
►►──────────────────────────────────────────►◄
      └─USSTAB=USS_definition_table_name─┘
```

```
          ┌─VERSALID=NO─────┐
►►────────┴─────────────────┴─►◄
      └─VERSALID=─┬─NO──┬─┘
                  └─YES─┘
```

```
►►────────────────────────────►◄
      └─VNGROUP=group_name─┘
```

```
►►──────────────────────────────────────────►◄
      └─VNNAME=connection_network_cpname─┘
```

```
►►──VTAMTOPO=─┬─NOREPORT─┬─►◄
              ├─REPORT───┤
              ├─NOSWPUS──┤
              └─NOLLINES─┘
```

Keyword Descriptions

symbol

```
►►───symbol───►◄
```

Use: Required
Default: None

Provides a name for the line group. *symbol* can be any valid assembler language
symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to
avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process
creates.

ACTIVTO

```
►►────────────────────────►◄
      └─ACTIVTO=seconds─┘
```

Specifies, in seconds, how long the secondary NCP is to wait for communication
from the primary NCP before initiating shutdown. You can specify this value as an
integer or with one decimal place.

Specify ACTIVTO for switched subarea lines on the configurable GROUP definition
statement if you specify BRKCON on the PU definition statement.

Table 2-23 on page 2-106 lists the default, minimum, and maximum values for ACTIVTO.

Table 2-23. Values for ACTIVTO on the GROUP Definition Statement

| | ACTIVTO Values | | |
Line Type	Default	Minimum	Maximum
Switched peripheral and nonswitched	60.0	1.0	420.0
Switched subarea	0.0	0.0	6553.5

The value of ACTIVTO for the secondary NCP must be greater than the value of PAUSE on the LINE definition statement for the primary NCP.

As a guide in determining the appropriate ACTIVTO value, choose a value such that the following expression is true:

$$(maxout - ipl) \geq count \geq (t \times n) \geq sdo$$

where:

maxout is the maximum time the controller can be out of service because of link failure.

ipl is the interval normally required to load (IPL) the controller. If loading is not required, the value of *ipl* can be 0.

count is the value of the ACTIVTO keyword.

t and *n* are the values defined on the RETRIES keyword on the PU definition statement of the primary NCP.

sdo is the average duration of *short-duration outages* of the subarea link (outages caused by momentary loss of modem carrier, excessive noise, power surges, or other disruptive line conditions).

ADAPTER
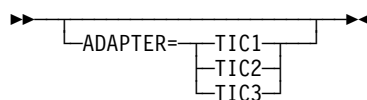
Use: Token-Ring physical resources
Default: None

Specifies which type of Token-Ring interface coupler (TIC) is attached to this group of physical lines.

TIC1│TIC2

Use: NTRI

Specifies whether the lines in this group are attached to a type 1 TIC or a type 2 TIC. Do not code ADAPTER=TIC1 if you code ECLTYPE=(PHYSICAL,ANY) on this GROUP definition statement because the type 1 TIC does not support a mixture of subarea and peripheral connections.

TIC3

Use: 3746 Model 900 Token Ring

Specifies that the lines in this group are attached to a type 3 TIC. Code ADAPTER=TIC3 to define a group of 3746 Model 900 Token-Ring physical lines.

The value of ADAPTER affects the default for ECLTYPE. See page 2-115 for more information.

AUTOGEN

```
►►──┬──────────────────────┬──►◄
    └─AUTOGEN=number─┘
```
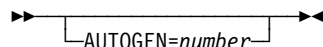
Use: Token-Ring logical peripheral lines
               Frame-relay logical peripheral lines
               ISDN logical subarea and peripheral lines
Default: None
Maximum: 3000

Specifies how many LINE and PU definition statements NDF is to add automatically to this GROUP definition statement.

NTRI Frame Handler Logical Lines: The AUTOGEN keyword is not valid.

CAEXIT
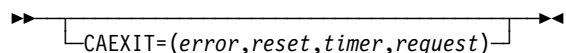
```
►►──┬──────────────────────────────────────┬──►◄
    └─CAEXIT=(error,reset,timer,request)─┘
```

Use: Channel user-written code

Specifies the entry points of user-written channel service routines. If user-written routines are to handle the channel operations, you must code all four entry points.

error

> Default: None

> Specifies the entry point of the user-written level 1 error recovery procedure routine.

reset

> Default: None

> Specifies the entry point of the user-written system reset routine.

timer

> Default: None

> Specifies the entry point of the user-written timer interrupt routine.

request

> Default: None

Specifies the entry point of the user-written routine that handles program request interrupts, suppress-out monitor requests, and permission-to-process stacked requests.

CHANLNK
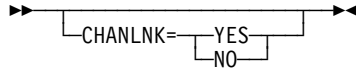
```
►►──┬──────────────────────┬──►◄
     └─CHANLNK=──┬──YES──┬──┘
                 └──NO───┘
```

Use: Channel user-written code
Default: None

Specifies whether this group defines channel adapters for which user-written routines handle emulator subchannel operations. CHANLNK is valid only if you code UCHAN=YES on the BUILD definition statement and LEVEL2=symbol, LEVEL5=NCP, LNCTL=SDLC, and VIRTUAL=NO on this GROUP definition statement.

If you specify UCHAN=YES on the BUILD definition statement, at least one line group with CHANLNK=YES is required.

If you specify CHANLNK, code CAEXIT on the GROUP definition statement, define the channel adapter logical address on ADDRESS on the LINE definition statement, define the subchannel address on LOCADDR on the LU definition statement, and code UCCB on the LU definition statement.

CHAREC

```
         ┌─CHAREC=XONOFF──────────────────────────────┐
►►───────┼────────────────────────────────────────────┼──►◄
         └─CHAREC=──┬──XONOFF───────────────────────┬──┘
                    │          ┌─XONOFF─┐            │
                    └──(──┬────┴────────┴────┬─,chars)┘
                          ├──XONOFF──┤
                          ├──XON─────┤
                          ├──XOFF────┤
                          └──NO──────┘
```
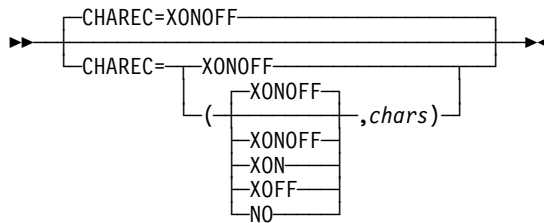
| Use: TWX lines (NCP or EP)
Default: XONOFF

Specifies the characters NCP or EP is to recognize as the end of a message when receiving data from a teletypewriter exchange service (TWX) terminal. When these characters are received from a line in emulation mode, NCP or EP signals channel-end and device-end status to the emulation subchannel associated with the line.

XONOFF
  Indicates either XON (DC1) or XOFF (DC2) and no other character.

(**XONOFF,**chars)
  Indicates XON, XOFF, or one or two user-defined characters.

(**XON,**chars)
  Indicates XON only, or one or two user-defined characters.

(**XOFF,**chars)
  Indicates XOFF only, or one or two user-defined characters.

(`NO,`*chars*)
> Indicates one or two user-defined characters, but not XON or XOFF.

Coding CHAREC=(,*chars*) is the same as coding CHAREC=(XONOFF,*chars*).

The program does not recognize any characters other than WRU, XON, XOFF, and EOT unless they are specified in CHAREC.

The characters specified must not be the reversed code for the characters desired as they were in previous releases. Specify *chars* as the parallel data field (PDF) representation of the characters the terminal sends. More information, including a discussion of ASCII character codes for TWX, is provided under "Line Character Codes" in the *NCP and EP Reference Summary and Data Areas*, Volume 2.

COMPACB

```
            ┌─COMPACB=NO──────┐
►►──────────┴─COMPACB─┬─YES─┬─┴──────►◄
                      └─NO──┘
```
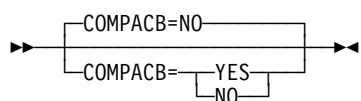
Use: IBM special products or user-written code
Default: NO

Specifies whether the user adapter control blocks (UACBs) for the lines in this group are compatible with the NCP ACBs for line trace, scanner interface trace, and wrap test.

Specify COMPACB=YES if the UACBs are closely compatible with NCP ACBs or COMPACB=NO if they are not. See the *NCP and SSP Customization Guide* for further information.

3746 Model 900 Token Ring: The COMPACB keyword is ignored.

3746 Model 900 Frame Relay: The COMPACB keyword is ignored.

Table 2-24 shows the effect on COMPACB when you code certain keywords on the GROUP definition statement.

Table 2-24. NDF Values for COMPACB on the GROUP Definition Statement

When You Code This on GROUP	The Effect on COMPACB Is
ECLTYPE	The only valid value is COMPACB=NO.
FRELAY=LOGICAL (3745 only)	The only valid value is COMPACB=NO.
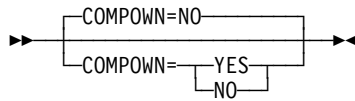
COMPOWN

```
                      ┌─COMPOWN=NO─────┐
►►────────────────────┴────────────────┴──────────►◄
          └─COMPOWN=──┬─YES─┬──┘
                      └─NO──┘
```

Use: IBM special products or user-written code
Default: NO

Specifies whether the control blocks for the lines in this group are compatible with the switchback, LA disconnect, and fallback functions.

If you specify COMPOWN=YES and the MOSS console requests a switchback, NCP verifies that IBM special products or user-written code lines installed on adapters to be switched have been freed by SSCP owners. However, if COMPOWN=YES and the MOSS operator forces a switchback, IBM special products or user-written code lines installed on switched adapters are deactivated if they are still active.

3746 Model 900 Token Ring: The COMPOWN keyword is ignored.

3746 Model 900 Frame Relay: The COMPOWN keyword is ignored.

Table 2-25 shows the effect on COMPOWN when you code certain keywords on the GROUP definition statement.

Table 2-25. NDF Values for COMPOWN on the GROUP Definition Statement

When You Code This on GROUP	The Effect on COMPOWN Is
ECLTYPE (NTRI only)	NDF adds COMPOWN=YES. The only valid value is COMPOWN=YES.
FRELAY=LOGICAL (3745 only)	NDF adds COMPOWN=YES. The only valid value is COMPOWN=YES.

COMPSWP
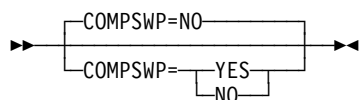
```
                    ┌─COMPSWP=NO─────┐
  ►►──────────────────────────────────────►◄
                    └─COMPSWP=──┬─YES─┬──┘
                               └─NO──┘
```

Use: IBM special products or user-written code
Default: NO

Specifies whether the control block structure for IBM special products or user-written code is compatible with the NCP control block structure for port swapping. If the control block structures are compatible, COMPSWP allows port swapping of lines controlled by IBM special products or user-written code. See the *NCP and SSP Customization Guide* for information about control block compatibility.

COMPSWP=YES indicates that this is a program using IBM special products or user-written code. When you specify COMPSWP=YES, NCP verifies that the IBM special products or user-written code and port swap requests are honored.

3746 Model 900 Token Ring: The COMPSWP keyword is ignored.

3746 Model 900 Frame Relay: The COMPSWP keyword is ignored.

Table 2-26 shows the effect on COMPSWP when you code certain keywords on the GROUP definition statement.

Table 2-26. NDF Values for COMPSWP on the GROUP Definition Statement

When You Code This on GROUP	The Effect on COMPSWP Is
ECLTYPE=PHYSICAL (NTRI only)	NDF adds COMPSWP=YES. The only valid value is COMPSWP=YES.
ECLTYPE=LOGICAL (NTRI only)	The only valid value is COMPSWP=NO.
FRELAY=LOGICAL (3745 only)	NDF adds COMPSWP=NO. The only valid value is COMPSWP=NO.
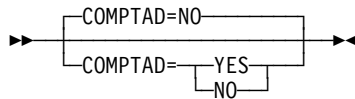
COMPTAD

```
>>─────────┬──COMPTAD=NO──────────┬──────><
           │                      │
           └──COMPTAD=──┬──YES──┬─┘
                        └──NO───┘
```

Use:	IBM special products or user-written code
Default:	NO

Specifies whether the control block structure of IBM special products or user-written code is compatible with NCP for initialization, configuration data set (CDS) update, bus switching initialization of the TA and TD bus address, and initialization of the AIT index. If you specify COMPTAD=NO, you must write your own code to accomplish these functions. See the *NCP and SSP Customization Guide* for more information.

3746 Model 900 Token Ring: The COMPTAD keyword is ignored.

3746 Model 900 Frame Relay: The COMPTAD keyword is ignored.

Table 2-27 shows the effect on COMPTAD when you code certain keywords on the GROUP definition statement.

Table 2-27. NDF Values for COMPTAD on the GROUP Definition Statement

When You Code This on GROUP	The Effect on COMPTAD Is
ECLTYPE=PHYSICAL (NTRI only)	NDF adds COMPTAD=YES. The only valid value is COMPTAD=YES.
ECLTYPE=LOGICAL (NTRI only)	The only valid value is COMPTAD=NO.
FRELAY=PHYSICAL (3745 only)	The default is COMPTAD=YES. The only valid value is COMPTAD=YES.
FRELAY=LOGICAL (3745 only)	The only valid value is COMPTAD=NO.

CRETRY
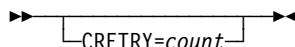
```
>>──────┬──────────────────┬──────><
        └──CRETRY=count─────┘
```

Use:	Start-stop and BSC lines in network control mode
Default:	See text
Minimum:	0
Maximum:	254

Specifies how many error recovery attempts NCP is to make when a recoverable error occurs on a line in this group. (Errors in control mode usually occur during polling or addressing.) If you code CRETRY=255, NCP retries the operation an unlimited number of times.

The default is CRETRY=2 for start-stop lines and CRETRY=7 for BSC lines.

DELAY

```
        ┌─DELAY=NO──────────┐
►►──────┤                   ├──────►◄
        └─DELAY=──┬─600──┬──┘
                  ├─1200─┤
                  └─NO───┘
```
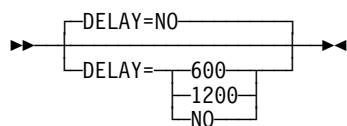
Use: WTTYs and start-stop lines in emulation mode
Default: NO
Minimum: 0.0
Maximum: See text

Specifies whether the World Trade Teletypewriter (WTTY) terminals (teleprinters) specified by this GROUP definition statement require a line turnaround time of 70 to 80 milliseconds and, if so, their operation speed in bits per second.

Code DELAY=600 if the terminals require the delayed turnaround and operate at 600 bps. Code DELAY=1200 if the terminals require the delayed turnaround and operate at 1200 bps.

If delayed turnaround is not required, specify DELAY=NO or omit the keyword.

Channel Links: If you code LNCTL=CA on this GROUP definition statement, DELAY has the same function as on the LINE definition statement. See page 2-217 for more information.

DIAL

```
        ┌─DIAL=NO───────┐
►►──────┤               ├──────►◄
        └─DIAL=──┬─YES─┬─┘
                 └─NO──┘
```
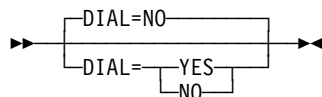
Use: NCP or EP
Default: NO

Specifies whether the lines in the group require switched line control procedures. If they do, code DIAL=YES. If they do not, code DIAL=NO.

You must code DIAL=YES and PUTYPE=4 on the GROUP definition statement if you are defining switched subarea support lines.

If you code DIAL=YES on this GROUP definition statement, PUTYPE=4 on the PU definition statement, and IPL=YES on the LINE definition statement for X.25, switched X.21, 3746 Model 900 SDLC, or ISDN lines, the definitions for these lines do not have to agree with the Link IPL Ports (LKP) definitions maintained on the MOSS disk.

For casual connection, specify DIAL=NO when MODE=SECONDARY on the GROUP definition statement.

Subarea Lines: For subarea lines, an autocall unit must be installed; you cannot establish a manually dialed connection.

3745 Frame-Relay Physical Lines: Starting with NCP V7R6, you can specify DIAL=YES. See the section on defining frame-relay physical lines in the *NCP, SSP, and EP Resource Definition Guide*.

| **3746 Model 900 Frame-Relay Physical Lines:**   Starting with NCP V7R7, you can
| specify DIAL=YES. See the section on defining frame-relay physical lines
| in the *NCP, SSP, and EP Resource Definition Guide*.

| **Emulation Mode:** DIAL=NO must be specified if a manual Data Access Arrange-
| ment (DAA) device is used and the call is manually originated and
| answered.

Table 2-28 shows the effect on DIAL when you code certain keywords on the
GROUP definition statement.

Table 2-28. NDF Values for DIAL on the GROUP Definition Statement

When You Code This on GROUP	The Effect on DIAL Is
ECLTYPE=PHYSICAL	NDF adds DIAL=NO. The only valid value is DIAL=NO.
ECLTYPE=(LOGICAL,PERIPHERAL)	NDF adds DIAL=YES. The only valid value is DIAL=YES.
ECLTYPE=(LOGICAL,SUBAREA) or ECLTYPE=(LOGICAL,FRELAY)	NDF adds DIAL=NO. The only valid value is DIAL=NO.
NCST=IP	NDF adds DIAL=NO. The only valid value is DIAL=NO.
ETHERNET=PHYSICAL	NDF adds DIAL=NO. The only valid value is DIAL=NO.
FRELAY=(PHYSICAL,ANY) FRELAY=LOGICAL FRELAY=(LOGICAL,SUBAREA)	NDF adds DIAL=NO. The only valid value is DIAL=NO.
FRELAY=(LOGICAL,PERIPHERAL)	NDF adds DIAL=YES. The only valid value is DIAL=YES.
ISDN=PHYSICAL	NDF adds DIAL=NO. The only valid value is DIAL=NO.
ISDN=(LOGICAL,B) ISDN=(LOGICAL,SUBAREA) ISDN=(LOGICAL,PERIPHERAL)	NDF adds DIAL=YES. The only valid value is DIAL=YES.

DSRTO

```
        ┌─DSRTO=51──────┐
►►──────┤               ├──►◄
        └─DSRTO=seconds─┘
```
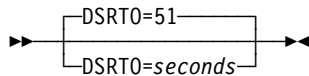
Use: Leased BSC lines in emulation mode
Default: 51
Minimum: 1
Maximum: 1632

| Specifies, in seconds, how long NCP or EP is to wait for an emulator line to
become DSR before timing out on an ENABLE command. This keyword is useful
when you use manually dialed modems to make connections.

ECLTYPE

```
►►──┬─────────────────────────────────────────────────────────────┬──►◄
     └─ECLTYPE=(──┬─PHYsical─┬──,──┬─PERipheral──┬──)─┘
                  └─LOGical──┘     ├─SUBarea─────┤
                                   ├─ANY──(1)────┤
                                   └─FRELAY──(2)─┘
```
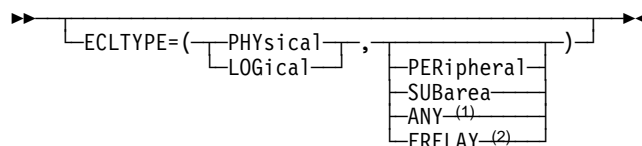
¹ Valid for physical lines only

² Valid for logical lines only

Use: Required for Token-Ring resources
Default: See text

Specifies that this GROUP definition statement defines a physical or logical connection to a Token Ring.

When you code ECLTYPE, code NEWDEFN=YES on the OPTIONS definition statement. When you do this, NDF automatically adds certain definition statements and keywords to the generation definition. For a description of these additions, see the *NCP, SSP, and EP Resource Definition Guide*.

When you code ECLTYPE, code at least one GROUP definition statement with ECLTYPE=PHYSICAL to define the Token-Ring interface coupler (TIC) attached to the communication controller. Also, code at least one GROUP definition statement with ECLTYPE=LOGICAL to define the devices attached to the Token Ring.

You may define physical and logical subarea lines, physical and logical peripheral lines, and NTRI frame handler logical lines. Each logical subarea or peripheral line must be the same type (subarea or peripheral) as the corresponding physical line. For NTRI frame handler logical lines, the *type* value specified for the physical line does not affect the use of the NTRI attachment for switching frame-relay frames.

Physical Lines: The type 1 TIC supports either subarea or peripheral connections, but not a mixture of both. Consequently, when you are using a type 1 TIC (ADAPTER=TIC1), only SUBAREA and PERIPHERAL are valid for the second suboperand of ECLTYPE. When ADAPTER=TIC1, the default for the second suboperand of ECLTYPE is PERIPHERAL.

The type 2 and type 3 TICs support subarea connections, peripheral connections, or a mixture of both. Consequently, when you are using a type 2 or type 3 TIC (ADAPTER=TIC2 or ADAPTER=TIC3), PERIPHERAL, SUBAREA, and ANY are valid for the second suboperand of ECLTYPE. When ADAPTER=TIC2 or ADAPTER=TIC3, the default for the second suboperand of ECLTYPE is ANY.

If the physical lines in this group will be used only for peripheral devices, code PERIPHERAL as the second suboperand of ECLTYPE.

Frame Relay: If a physical line in this group will be used to support a frame-relay connection, ADAPTER=TIC1 or ADAPTER=TIC2 is the only valid value. NTRI physical lines supporting frame-relay traffic can support any other valid combination of physical connections.

Spare or Redefinable Lines: The ECLTYPE keyword is not valid.

Logical Lines: Code ECLTYPE=(LOGICAL,PERIPHERAL) to define a logical peripheral resource. If a logical peripheral resource is mapped to a physical line (using the PHYPORT and PHYSRSC keywords), then code ECLTYPE=(PHYSICAL,PERIPHERAL) or ECLTYPE=(PHYSICAL,ANY) for that physical line.

Code ECLTYPE=(LOGICAL,SUBAREA) to define a logical subarea resource. Each logical subarea resource must be mapped to a physical line defined as ECLTYPE=(PHYSICAL,SUBAREA) or ECLTYPE=(PHYSICAL,ANY). See PORTADD on the LINE definition statement on page 2-263 for more information.

Code ECLTYPE=(LOGICAL,FRELAY) to define a NTRI frame handler logical line group. ECLTYPE=(LOGICAL,FRELAY) is not valid when ADAPTER=TIC3 (ODLC) on the associated Token-Ring physical GROUP statement.

EOB
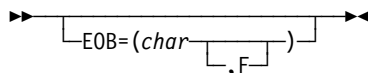
```
►►──────┬─────────────────────┬──────►◄
        └─EOB=(char──────┬───)─┘
                         └─,F─┘
```

Use: WTTY lines in emulation mode
Default: None

Specifies the character sequence that NCP or EP is to recognize as the EOB or end-of-message sequence when received from any WTTY terminal or certain U.S. and Canadian teletypewriter terminals. These terminals typically send either a sequence of 4 identical characters or a sequence of FIGS *char* LTRS.

If the terminal transmits a 4-character sequence, specify the character used as EOB=*char*, where *char* is the hexadecimal representation of the character transmitted. For example, if the terminal sends the sequence MMMM (in letters shift), code EOB=07.

If the terminal transmits a FIGS *char* LTRS sequence, code *char* as the hexadecimal representation of the character sent and also code the *F* following the character. For example, FIGS M LTRS is coded as EOB=(27,F).

| The "Line Character Codes" section of *NCP and EP Reference Summary and Data Areas*, Volume 2, lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

The EOB sequence can be specified in either the EOB or the CHAREC keyword, but not in both.

EOT
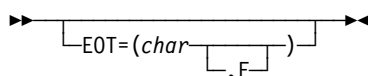
```
►►──────┬─────────────────────┬──────►◄
        └─EOT=(char──────┬───)─┘
                         └─,F─┘
```

Use: WTTY lines in emulation mode
Default: None

Specifies the character sequence that NCP or EP is to recognize as the EOT sequence when received from any WTTY terminal or certain U.S. and Canadian teletypewriter terminals. These terminals typically send, as an EOT indication, either a sequence of 4 identical characters or a sequence of FIGS *char* LTRS.

If the terminal transmits a 4-character sequence, specify the character used as EOB=*char*, where *char* is the hexadecimal representation of the character transmitted. For example, if the terminal sends the sequence AAAA (in letters shift), code EOT=18.

If the terminal transmits a FIGS *char* LTRS sequence, code *char* as the hexadecimal representation of the character sent and code the *F* following the character. For example, FIGS A LTRS is coded as EOT=(38,F).

The standard teletypewriter ending sequence is FIGS H LTRS. If you omit EOT, FIGS H LTRS is the sequence NCP or EP recognizes as the EOT sequence when receiving from a teletypewriter terminal.

The "Line Character Codes" section of *NCP and EP Reference Summary and Data Areas*, Volume 2, lists the transmission code bit patterns for the ITA2 and ZSC3 codes.

ETHERNET

```
►►──┬──────────────────────┬──►◄
    └─ETHERNET=PHYsical─────┘
```
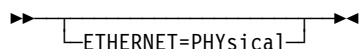
Use: Required for physical connections for Ethernet-type LANs

Specifies that the lines in this group are connected to an Ethernet-type LAN.

When you code ETHERNET, code NEWDEFN=YES on the OPTIONS definition statement.

Spare or Redefinable Lines: The ETHERNET keyword is not valid.

FRAMING

```
        ┌─FRAMING=11──────┐
►►──────┼─────────────────┼──►◄
        └─FRAMING=─┬─10─┬─┘
                   └─11─┘
```
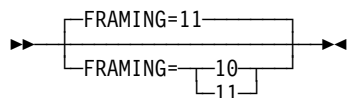
Use: TWX lines in network control mode
Default: 11

Specifies how many stop bits NCP is to send for each character and expects to receive from TWX terminals in this group. A value of 11 specifies that 2 stop bits are used. A value of 10 specifies that only 1 stop bit is used. Each TWX character comprises 1 start bit, 7 data bits, 1 parity bit, and 1 or 2 stop bits.

FRELAY

```
►►─┬──────────────────────────────────────────────┬─►◄
   └─FRELAY=(─┬─PHYsical─┬─,─┬────────────┬─)─┘
             └─LOGical──┘   ├─PERipheral─┤
                            ├─SUBarea────┤
                            ├─ANY──(1)───┤
                            └─IP─(2)─────┘
```
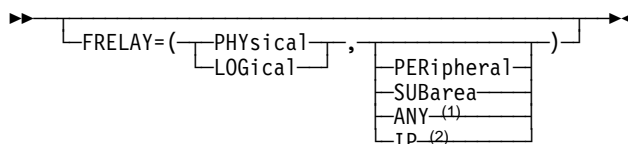
[1] Valid for physical lines only

[2] Valid for 3745 logical lines only

Use: Required for frame-relay resources
Default: See text

Specifies that this GROUP definition statement defines a frame-relay physical or logical line group.

When you code FRELAY, code NEWDEFN=YES on the OPTIONS definition statement. When you do this, NDF automatically adds certain definition statements and keywords to the generation definition. For a description of these additions, see the *NCP, SSP, and EP Resource Definition Guide*.

Physical Lines: The types of logical lines supported by a physical line is determined by the second suboperand of FRELAY. Frame-relay physical lines support frame-handler subports and frame-relay internet hosts regardless of the value specified for the second suboperand.

- Code FRELAY=(PHYSICAL,PERIPHERAL) to define a group of physical lines that support logical peripheral lines.

- Code FRELAY=(PHYSICAL,SUBAREA) to define a group of physical lines that support logical subarea lines.

- Code FRELAY=(PHYSICAL,ANY) to define a group of physical lines that support logical peripheral lines and logical subarea lines. FRELAY=(PHYSICAL,ANY) is valid only when ADDRESS specifies a relative line number in the range 1024–1039 (for an HPTSS line) or 2112–2750 (for a 3746 Model 900 frame-relay line).

3746 Model 900: The default for the second suboperand is ANY. This is valid only for NCP V7R2 and later releases.

HPTSS Adapters: If you code FRELAY=(PHYSICAL,ANY) and the microcode does not support subarea and peripheral connections on a single physical line, only subarea connections are supported. Without the correct microcode level, either subarea or peripheral connections can be supported on a single physical line, but not both. The HPTSS adapter must be loaded with one of the following microcode levels to support both peripheral and subarea connections.

- *C38006* for the IBM 3745 Models 130, 150, 160, 170
- *C37967* for the IBM 3745 Models 210, 310, 410, 610
- *D39888* for the IBM 3745 Models 21A, 31A, 41A, 61A
- *D400002* for the IBM 3745 Model 17A

If you code FRELAY=PHYSICAL and do not specify SUBAREA, PERIPHERAL, or ANY, the default for the second suboperand depends on the line address specified by ADDRESS on the LINE definition statement. If the line address is 0–895, the default is FRELAY=(PHYSICAL,SUBAREA). If the

line address is 1024–1039 or 2112–2750, the default is
FRELAY=(PHYSICAL,ANY).

Logical Lines: Code FRELAY=LOGICAL or FRELAY=(LOGICAL,SUBAREA) to
define a logical subarea line group. Code
FRELAY=(LOGICAL,PERIPHERAL) to define a logical peripheral line group.
| Code FRELAY=(LOGICAL,IP) to define a logical IP line group.

If you code FRELAY=LOGICAL or FRELAY=(LOGICAL,SUBAREA), the PU
definition statement specified by the PHYSRSC keyword must be included
in a line group with FRELAY=PHYSICAL,
FRELAY=(PHYSICAL,SUBAREA), or FRELAY=(PHYSICAL,ANY) on the
GROUP definition statement.

If you code FRELAY=(LOGICAL,PERIPHERAL) and you do not code
PHYSRSC=NONE, the PU definition statement specified by the PHYSRSC
keyword must be included in a line group with
FRELAY=(PHYSICAL,PERIPHERAL) or FRELAY=(PHYSICAL,ANY) on the
GROUP definition statement.

| IP support is defined as a logical group only for frame-relay switched phys-
| ical lines. If you code FRELAY=(LOGICAL,IP), the physical PU definition
| statement specified by the PHYSRSC keyword must be under a GROUP
| statement defined as FRELAY=PHYSICAL and DIAL=YES. Only one
| logical LINE statement can be defined under the logical GROUP.

| **3746 Model 900:** For logical lines, SUBAREA and PERIPHERAL are valid
| for the second suboperand, but IP is not.

Spare or Redefinable Lines: The FRELAY keyword is not valid.

ISDN

```
►►──┬──────────────────────────────────────────────────────┬──►◄
    └─ISDN=─┬──(─LOGical──┬──,B─────────┬──)──┘
            │            ├──,SUBarea───┤
            │            └──,PERipheral┘
            └─PHYsical────────────────────────┘
```
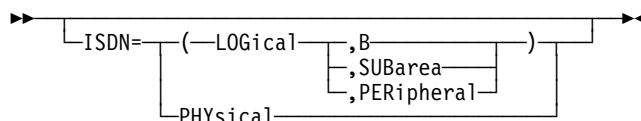
Use: Required for ISDN resources

Specifies that this GROUP definition statement defines an ISDN physical or logical
line group.

When you code ISDN, code NEWDEFN=YES on the OPTIONS definition state-
ment. When you do this, NDF automatically adds certain definition statements and
keywords to the generation definition. For a description of these additions, see the
NCP, SSP, and EP Resource Definition Guide.

ISDN Physical Lines: Code ISDN=PHYSICAL to define an ISDN physical line (D
channel).

ISDN B-Channels: Code ISDN=(LOGICAL,B) to define an ISDN B-channel.
Although an ISDN B-channel is a logical entity, it acts similarly to a frame-
relay physical line with ISDN resources multiplexed over it. In this book, a
reference to ISDN logical lines does not include ISDN B-channels.

ISDN Logical Lines: Code ISDN=(LOGICAL,SUBAREA) to define an ISDN logical subarea line. Code ISDN=(LOGICAL,PERIPHERAL) to define an ISDN logical peripheral line.

Spare or Redefinable Lines: The ISDN keyword is not valid.

KBDLOCK

```
             ┌─KBDLOCK=80─┐
►►───────────┤            ├───►◄
             └─KBDLOCK=char─┘
```
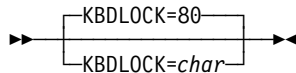
Use: TWX lines in network control mode
Default: 80

Specifies, in hexadecimal, the control character NCP is to send to TWX terminals to signal operators to stop using the keyboard. This technique, called logical keyboard lock, stops data entry by the terminal operator before NCP is ready to receive. The signal is sent only when the TSO/TCAM monitor mode is active for the line.

If you omit KBDLOCK or code KBDLOCK=80, NCP does not send any keyboard lock characters. If the null is sent, the line is put in a mark state (X'FF' is sent with the secondary control field [SCF] pad flag on).

KBDLOCK is valid only if you code LNCTL=SS on the GROUP definition statement, MONITOR=YES and POLLED=NO on the LINE definition statement, and TERM=TWX on the TERMINAL (or higher-level) definition statement.

LEVEL2
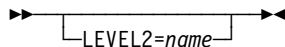
```
►►───────────────────►◄
   └─LEVEL2=name─┘
```

Use: User-written line control, NTRI, and 3745 frame relay
Default: None

Specifies the entry point of the user-written, NTRI, or 3745 frame-relay code for level 2 scanner interrupts. LEVEL2 is required if you define LNCTL=USER. Level 2 interrupt code is optional if LNCTL=SS, BSC, or SDLC.

If you code LEVEL2, also code the LEVEL3 keyword.

3746 Model 900 Token Ring: The LEVEL2 keyword is ignored.

3746 Model 900 Frame Relay: The LEVEL2 keyword is ignored.

Table 2-29 on page 2-121 shows the effect on LEVEL2 when you code certain keywords on the GROUP definition statement.

Table 2-29. NDF Values for LEVEL2 on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LEVEL2 Is
ECLTYPE=PHYSICAL (NTRI only)	NDF adds LEVEL2=ECLNARL2. The only valid value is LEVEL2=ECLNARL2.
ECLTYPE=LOGICAL (NTRI only)	NDF adds LEVEL2=ECLNAVL2. The only valid value is LEVEL2=ECLNAVL2.
FRELAY=LOGICAL (3745 only)	NDF adds LEVEL2=ECLNAVL2. The only valid value is LEVEL2=ECLNAVL2.

LEVEL3
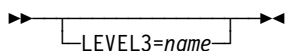
```
►►──┬──────────────┬──►◄
    └─LEVEL3=name──┘
```

Use: User-written line control, NTRI, and 3745 frame relay
Default: None

Specifies the entry point of the user-written, NTRI, or 3745 frame-relay code for level 3 interrupts. LEVEL3 is required if you specify LEVEL2.

3746 Model 900 Token Ring: The LEVEL3 keyword is ignored.

3746 Model 900 Frame Relay: The LEVEL3 keyword is ignored.

Table 2-30 shows the effect on LEVEL3 when you code certain keywords on the GROUP definition statement.

Table 2-30. NDF Values for LEVEL3 on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LEVEL3 Is
ECLTYPE=PHYSICAL (NTRI only)	NDF adds LEVEL3=ECLNARL3. The only valid value is LEVEL3=ECLNARL3.
ECLTYPE=LOGICAL (NTRI only)	NDF adds LEVEL3=ECLNAVL3. The only valid value is LEVEL3=ECLNAVL3.
FRELAY=LOGICAL (3745 only)	NDF adds LEVEL3=ECLNAVL3. The only valid value is LEVEL3=ECLNAVL3.

LEVEL5

```
                    ┌─LEVEL5=USER─┐
►►──────────────────┴────────────┴───────────────►◄
              └─LEVEL5=─┬─NCP──┐
                        └─USER─┘
```
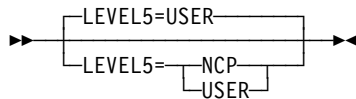
Use: User-written line control, NTRI, and 3745 frame relay
Default: USER

Specifies whether the level 5 code is provided by NCP or is user-written. LEVEL5 is valid only if you specify LEVEL2 and LEVEL3. If you omit this keyword and define LEVEL2 and LEVEL3, LEVEL5=USER is the default.

LEVEL5=NCP specifies that the level 5 code is provided by NCP. In this case, LNCTL must equal SS, BSC, or SDLC. The COMP, CLUSTER, LINE, LU, PU, and TERMINAL definition statements also must follow the same requirements as the line groups supported totally by NCP-supplied code.

LEVEL5=USER specifies that the level 5 code is user-written and can be specified for any type of line control. When you select USER, the NCP generation process assumes that level 5 code is supplied separately; therefore, no level 5 control blocks are generated for this group.

If LEVEL5=USER and VIRTUAL=NO, only the LINE definition statement can be specified on this GROUP. You must include a separate GROUP definition statement with VIRTUAL=YES to attach the lines to the network. If VIRTUAL=YES on GROUP, the LINE, LU, PU, and SERVICE definition statements must follow the same requirements as the line groups supported by NCP-supplied code.

3746 Model 900 Token Ring: The LEVEL5 keyword is ignored.

3746 Model 900 Frame Relay: The LEVEL5 keyword is ignored.

Table 2-31 shows the effect on LEVEL5 when you code certain keywords on the GROUP definition statement.

Table 2-31. NDF Values for LEVEL5 on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LEVEL5 Is
ECLTYPE (NTRI only)	NDF adds LEVEL5=NCP. The only valid value is LEVEL5=NCP.
FRELAY=LOGICAL (3745 only)	NDF adds LEVEL5=NCP. The only valid value is LEVEL5=NCP.

LINEADD
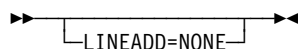
```
►►─────────────────────────►◄
      └─LINEADD=NONE─┘
```

Use: User-written line control, NTRI, and 3745 frame relay
Default: None

Specifies that no physical address interface is defined for this line group. LINEADD
is valid only if you code LEVEL2 and VIRTUAL=NO on this GROUP definition
statement. If you code LINEADD, do not code ADDRESS on the LINE definition
statement.

3746 Model 900 Token Ring: If you code LINEADD, it is deleted by NDF.

3746 Model 900 Frame Relay: The LINEADD keyword is ignored.

Table 2-32 shows the effect on LINEADD when you code certain keywords on the
GROUP definition statement.

Table   2-32. NDF Values for LINEADD on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LINEADD Is
ECLTYPE=LOGICAL (NTRI only)	NDF adds LINEADD=NONE.
FRELAY=LOGICAL (3745 only)	NDF adds LINEADD=NONE.

LINEAUT

```
►►─────────────────────────►◄
      └─LINEAUT=YES─┘
```
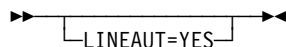
Use: User-written line control, Token-Ring, ISDN, and frame-relay
            resources
Default: None

Specifies auto-call for a line group for which no physical address interface is
defined. LINEAUT is valid only if you code LINEADD=NONE and DIAL=YES. Do
not specify the AUTO keyword on the LINE definition statement.

Table 2-33 shows the effect on LINEAUT when you code certain keywords on the
GROUP definition statement.

Table   2-33. NDF Values for LINEAUT on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LINEAUT Is
ECLTYPE=(LOGICAL,PERIPHERAL)	NDF adds LINEAUT=YES.
FRELAY=(LOGICAL,PERIPHERAL)	NDF adds LINEAUT=YES.
ISDN=(LOGICAL,SUBAREA) ISDN=(LOGICAL,PERIPHERAL)	NDF adds LINEAUT=YES.
DIAL=YES and LINEADD=NONE	The default is LINEAUT=YES.

LNCTL

```
►►──────┬─────────────────────┬──────►◄
        └─LNCTL=──┬─BSC──┬─────┘
                  ├─CA───┤
                  ├─SDLC─┤
                  ├─SS───┤
                  └─USER─┘
```
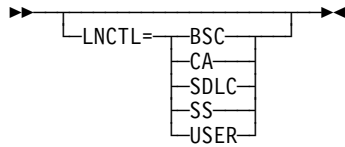
| Use: | NCP or EP |
| Default: | None |

Specifies the type of line control used for the lines in this group. If you omit LNCTL from this GROUP definition statement, the following assumptions are made about the lines in this group:

- If the preceding GROUP definition statement specifies LNCTL=SDLC, the lines in this group are also SDLC.

- If the preceding GROUP definition statement does *not* specify LNCTL=SDLC, the lines in this group are start-stop.

- If the preceding GROUP definition statement specifies LNCTL=CA, the lines in this group are channel links.

- If ECLTYPE is coded on this GROUP definition statement, the lines in this group are Token-Ring lines.

LNCTL=SDLC is valid for a line that is operated only in network control mode. (Specify TYPE=NCP on the GROUP definition statement.)

If LNCTL=SS, BSC, or SDLC, the LEVEL5 keyword specifies whether NCP or the user supplies the level 5 support.

Code LNCTL=SS or LNCTL=BSC when TYPE=EP and you code V25BIS on the LINE definition statement.

Code LNCTL=CA to define a group of channel links. GROUP definition statements for channel links must follow all GROUP definition statements for SDLC lines in the generation definition.

Code LNCTL=USER to define a group of lines with user-written line control. GROUP definition statements for user lines must follow all GROUP definition statements for SDLC lines in the generation definition. Also code LEVEL2, LEVEL3, and LEVEL5 for such groups. LEVEL5 specifies whether level 5 code is provided by NCP or is user-written. If level 5 code is user-written, only the GROUP, LINE, and SERVICE definition statements can be coded for this group.

ESCON Links: Code LNCTL=CA to define ESCON links.

Token Ring: Token-Ring lines use IBM Token-Ring protocols that are compatible with the Institute of Electrical and Electronic Engineers (IEEE) 802.5 standard.

Spare or Redefinable Lines: Only LNCTL=SDLC is valid.

Table 2-34 on page 2-125 shows the effect on LNCTL when you code certain keywords on the GROUP definition statement.

Table 2-34. NDF Values for LNCTL on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LNCTL Is
ECLTYPE	NDF adds LNCTL=SDLC. The only valid value is LNCTL=SDLC.
NCST=IP	NDF adds LNCTL=SDLC. The only valid value is LNCTL=SDLC.
ETHERNET=PHYSICAL	NDF adds LNCTL=SDLC. The only valid value is LNCTL=SDLC.
FRELAY	NDF adds LNCTL=SDLC. The only valid value is LNCTL=SDLC.
ISDN	NDF adds LNCTL=SDLC. The only valid value is LNCTL=SDLC.

LNKOWNER
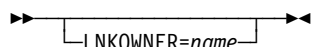
```
►►──┬──────────────────┬──►◄
    └─LNKOWNER=name─────┘
```

Default: None

Specifies the name of the user-written generation load module that is responsible for the link-level processing of the resources defined by this GROUP definition statement. *name* must match the name of one of the user-written generation load modules specified by USERGEN on the OPTIONS definition statement. The start record of this load module's statement and keyword vector table (SKVT) must indicate that the load module is valid for LNKOWNER.

LNKOWNER is valid only if you code LEVEL2 on this GROUP definition statement.
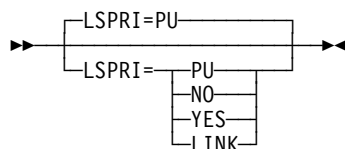
LSPRI

```
          ┌─LSPRI=PU──────────┐
►►──┬──────────────────────┬──►◄
    └─LSPRI=──┬─PU──┬───────┘
             ├─NO──┤
             ├─YES─┤
             └─LINK┘
```

Use: SDLC lines in network control mode; Ethernet-type, Token-Ring, ISDN, and frame-relay lines
Default: See text

Specifies whether transmission priority is to be used to determine the order in which PIUs are transmitted to the physical unit. Do not code LSPRI if you specify VIRTUAL=YES on the GROUP definition statement.

The default is LSPRI=PU for SDLC lines.

NO
Specifies that transmission priority is not to be used as the basis of transmission order. NCP transmits PIUs in the order in which they are received.

YES
Enables user-written code to prioritize PIUs. LSPRI=YES is valid only for lines on which you specify LEVEL2.

PU

Specifies that the link outbound queue (LOBQ) will be prioritized for each phys-
ical unit on the line. The priority of sessions across the physical unit is based
on the transmission priority of virtual routes associated with the sessions.
LSPRI=PU is valid only if you specify LNCTL=SDLC for peripheral lines,
LNCTL=USER, or LEVEL2 on the GROUP definition statement.

LINK

Replaces the service order table (SOT) with a prioritized queue used for trans-
mission. The SOT is still used for polling. You may code LSPRI=LINK only for
type 1 or type 2 physical units on either duplex lines or half-duplex lines on
which HDXSP is specified. LSPRI=LINK is valid only for peripheral SDLC lines
or IBM special products or user-written code.

Note: Coding LSPRI=LINK causes transmission priority to become the trans-
mission order basis not only across the physical unit but for *all* PIUs
across the link.

Token Ring: For Token-Ring physical lines, if you code
ECLTYPE=(PHYSICAL,PERIPHERAL) or ECLTYPE=(PHYSICAL,ANY), you
can code LSPRI=NO or LSPRI=PU. If you code
ECLTYPE=(PHYSICAL,SUBAREA), only LSPRI=NO is valid.

For Token-Ring logical lines, only LSPRI=NO is valid.

Frame Relay: For frame-relay physical lines, if you specify
FRELAY=(PHYSICAL,PERIPHERAL) or FRELAY=(PHYSICAL,ANY), you
can code LSPRI=PU or LSPRI=NO. If you specify
FRELAY=(PHYSICAL,SUBAREA), only LSPRI=NO is valid.

For frame-relay logical lines, only LSPRI=NO is valid. The default is
LSPRI=NO.

ISDN: For ISDN physical lines (ISDN=PHYSICAL), you can specify either
LSPRI=NO (the default) or LSPRI=PU.

Table 2-35 shows the effect on LSPRI when you code certain keywords on the
GROUP definition statement.

Table 2-35. NDF Values for LSPRI on the GROUP Definition Statement

When You Code This on GROUP	The Effect on LSPRI Is
LEVEL2	The default is LSPRI=NO.
ECLTYPE	The default is LSPRI=NO.
FRELAY=LOGICAL	The default is LSPRI=NO.

MODE

```
►►──┬──────────────────────────────┬──►◄
    └─MODE=─┬─PRImary───┬───────────┘
           └─SECondary─┘
```
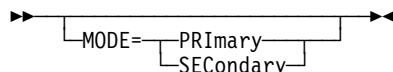
Default: None

Specifies the mode of operation of a line group.

Code MODE=PRIMARY on the GROUP definition statement for the NCP that will be primary, and code MODE=SECONDARY on the GROUP definition statement for the NCP that will be secondary.

For peripheral links, the default is MODE=PRIMARY.

For subarea links, the MODE keywords in both the GROUP and the associated SDLCST definition statement must be coded the same. You can specify the primary mode as PRIMARY and the secondary mode as SECONDARY. MODE is not valid if you code VIRTUAL=YES on this GROUP definition statement. Multi-point subarea links do not require an SDLCST definition statement or the SDLCST keyword.

NTRI Frame Handler Logical Lines: The MODE keyword is not valid.

NCST
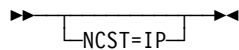
```
►►──┬───────────┬──►◄
    └─NCST=IP───┘
```

Use: Internet Protocol
Default: None

Specifies that the logical units in this line group support Internet Protocol using NCP connectionless SNA transport (NCST). Only one line group within an NCP can specify NCST=IP. If you specify NCST=IP, you must also specify NEWDEFN=YES on the OPTIONS definition statement and you must code the REMLU keyword on the LU definition statement.

Spare or Redefinable Lines: The NCST keyword is not valid.

NETID

```
►►──┬─────────────────────────────────┬──►◄
    └─NETID=(name─┬────────┬─)─────────┘
                  └─,ANY───┘
```
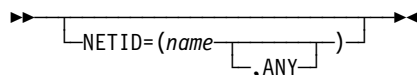
Use: Gateway NCP
Default: None

Specifies the name of the network associated with this group of subarea lines.

name
    Specifies either the native or non-native network (NETID on the BUILD or NETWORK definition statement) associated with the subarea line group when defining a subarea line group (PUTYPE=4). For more information, see NETID on the PU definition statement on page 2-400.

ANY

Specifies that NCP is to negotiate the name of the adjacent network during link activation. If you specify NETID=(*name*,ANY), the suboperand ANY is applied to all of the physical units defined in this group, and *name* becomes the default NETID for this group. You can code NETID on the PU definition statement to override the default name, but you cannot override the ANY suboperand. NETID=(*name*,ANY) is valid only if you code LNCTL=SDLC.

When NETID=(*name*,ANY) is specified for both ends of a nonswitched link, NCP uses the first *name* in EBCDIC order. *name* is also used for the connection to a migration node that does not send a NETID during XID exchange.

For dynamic network identification, specify NETID=(*name*,ANY) for subarea lines to allow any NETID to be received on this line, regardless of whether it is defined to NCP. See COPIES on the NETWORK definition statement on page 2-332 for information about model networks.

NPARSC

```
                ┌─NPARSC=NO───────┐
►►──────────────┤                 ├──►◄
                └─NPARSC=──┬─YES─┬─┘
                          └─NO──┘
```
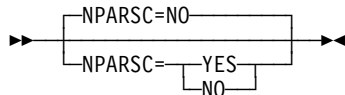
Use: NetView Performance Monitor (NPM) data collection
Default: NO

Specifies that this line group is a network performance analyzer resource used to send and receive NPM data and commands between NCP and the host. Only one line group within an NCP can specify NPARSC=YES.

NPARSC is valid only if you code NPA=YES on the BUILD definition statement. Coding NPA=YES also means that NPARSC is required on the GROUP definition statement that does NPA data collection.

If this GROUP definition statement is for NPA, code VIRTUAL=YES and LNCTL=SDLC, and omit CHANLNK. All other GROUP definition statement keywords are ignored.

Spare or Redefinable Lines: The NPARSC keyword is not valid.

PADCNT
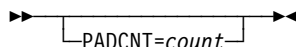
```
►►──────────────────────►◄
    └─PADCNT=count─┘
```

Use: WTTY lines in network control mode
Default: None
Minimum: 0
Maximum: 255

Specifies how many idle characters NCP is to send to a World Trade Teletypewriter (WTTY) terminal in this group to permit its motor to reach full speed before receiving data. PADCNT is required only for a line where terminals are not equipped with a continuously running motor. Specify a sufficient number of characters to create a 1.5-second delay on the line. The characters are sent only when the line is in network control mode.

PECHAR

```
    ┌─PECHAR=FF─┐
►►──┴───────────┴──────────────►◄
    └─PECHAR=─┬─FF───┬─┘
             └─char─┘
```
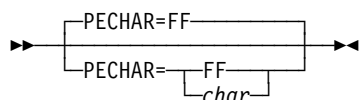
Use: TWX lines in network control mode
Default: FF

Specifies the hexadecimal representation of the data character that NCP uses to overlay data received from a TWX terminal when the program detects incorrect parity.

This keyword is valid only if you code PARCHK=ODD or PARCHK=EVEN on one or more LINE definition statements in this group.

PHYPORT

```
►►──┬──────────────────────┬──►◄
    └─PHYPORT=─┬─number─┬─┘
              └─NONE───┘
```
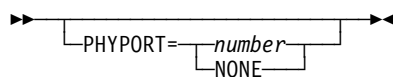
Use: NTRI logical lines
Default: None
Minimum: 0
Maximum: 99

Associates the logical lines in this group with a particular physical line. To do this, code the value specified by PORTADD on the LINE definition statement for the physical line. PHYPORT is required if this group includes logical subarea lines and you do not code PHYSRSC on this GROUP definition statement.

If you code PHYPORT=NONE, the lines in this group are not associated with any particular physical line. Instead, NCP dynamically associates these logical lines with physical lines as needed. You can do this only for peripheral lines; do not code PHYPORT=NONE if this group includes subarea lines.

Note: You may want to use the PHYSRSC keyword instead of PHYPORT. PHYSRSC associates a group of logical lines with the link station for a particular physical line. In addition, PHYSRSC causes VTAM to defer the activation of logical subarea lines until the station is activated.

NTRI Frame Handler Logical Lines: Code the PHYSRSC keyword to associate this group of logical lines with a physical station. The PHYPORT keyword is ignored.

3746 Model 900 Token Ring: Code the PHYSRSC keyword to associate this group of logical lines with a physical station. The PHYPORT keyword is ignored.

PHYSRSC
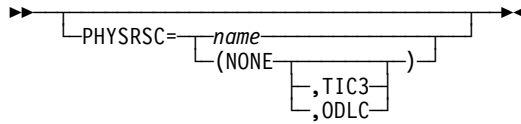
```
►►──┬─────────────────────────────────────────┬──►◄
    └─PHYSRSC=─┬─name────────────────────┬─┘
              └─(NONE─┬──────────┬─)─┘
                     ├─,TIC3─┤
                     └─,ODLC─┘
```

Use: Token-Ring, frame-relay, ISDN, and ESCON logical lines

Associates the logical lines in this group with the link station for a particular physical line. To do this, code PHYSRSC=*name*, where *name* is the name of the PU definition statement that defines the link station of the physical line.

Token Ring: Code PHYSRSC=NONE and ECLTYPE=(LOGICAL,PERIPHERAL) to define a pool of logical peripheral lines that NCP can dynamically associate with Token-Ring physical lines. If you code PHYSRSC=(NONE,TIC3) (NCP V6R2 or later) or PHYSRSC=(NONE,ODLC) (NCP V7R2 or later), the logical peripheral lines can be used only with 3746 Model 900 Token-Ring lines. If TIC3 or ODLC is omitted, the logical peripheral lines can be used only with NTRI connections.

When ECLTYPE=(LOGICAL,SUBAREA), the name specified on PHYSRSC must be the physical PU definition statement with NETWORK=SNA defined on a NTRI physical line. VTAM defers the activation of those lines until the SNA physical station is activated.

When ECLTYPE=(LOGICAL,FRELAY), the name specified on PHYSRSC must be the physical PU definition statement with NETWORK=FRELAY defined on a NTRI physical line.

3746 Model 900 Token Ring: PHYSRSC is required for 3746 Model 900 Token-Ring logical lines.

Frame-Relay Logical IP Lines: PHYSRSC is required for frame-relay logical IP lines. Code PHYSRSC=*name*; PHYSRSC=NONE is not valid. The PU definition statement identified by *name* must precede this GROUP definition statement and must be the first PU definition statement following the LINE definition statement for the frame-relay physical line. Other IP logical line groups cannot use the same PHYSRSC *name*.

Note: IP support is defined as a logical group only for frame-relay switched physical lines.

Frame-Relay Logical Subarea Lines: PHYSRSC is required for frame-relay logical subarea lines. Code PHYSRSC=*name*; PHYSRSC=NONE is not valid. The PU definition statement identified by *name* must precede this GROUP definition statement and must be the first PU definition statement following the LINE definition statement for the frame-relay physical line.

Frame-Relay Logical Peripheral Lines: Code PHYSRSC=*name* to explicitly associate a group of logical peripheral lines with a frame-relay physical line. Code PHYSRSC=NONE to define a group of unassigned logical peripheral lines. PHYSRSC=NONE resources are not subordinate resources of the physical line.

3746 Model 900 Frame Relay: PHYSRSC=(NONE,ODLC) defines a pool of 3746 Model 900 logical lines. If FRELAY=LOGICAL was specified,

PHYSRSC=(NONE,ODLC) or PHYSRSC=*name* is required. (*name* is the name of the first PU coded under a frame-relay physical line.)

ISDN B-Channels: PHYSRSC is required for ISDN B-channels. Code PHYSRSC=*name*; to associate a group of ISDN B-channels with the link station for a particular physical line. PHYSRSC=NONE is not valid. The PU definition statement identified by *name* must precede the GROUP definition statement.

ISDN Logical Subarea and Peripheral Lines: Only PHYSRSC=NONE is valid. It defines a pool of ISDN logical lines. If you do not code PHYSRSC, NDF adds it.

ESCON Links: PHYSRSC is required for ESCON logical links. Code PHYSRSC=*name*; PHYSRSC=NONE is not valid.

PRODID

```
►►─┬─────────────────┬─►◄
   └─PRODID=name─────┘
```
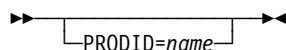
Use: IBM special products or user-written line control
Default: None

Specifies the name of the product identifier (subvector X'11') defined for the IBM special product or user-written code.

The IBM special products or user-written code must include the following fields in subvector X'11':

1. The length of subvector X'11' in the first byte

2. X'11' in the second byte

3. The identifier for the IBM special product or user-written code in the remainder of the field.

If PRODID is specified, subvector X'11' will appear in the generic alert generated for a dial-out failure.

PUTYPE

```
►►─┬─PUTYPE=─┬─1─────┬─┬─►◄
             ├─2─────┤
             ├─(1,2)─┤
             ├─4─────┤
             └─5─────┘
```
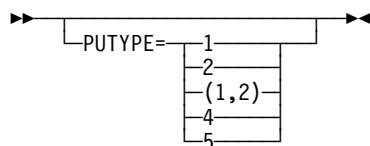
Default: None

Specifies the physical unit type of the SDLC device you are defining. You must code PUTYPE=4 on the GROUP definition statement for switched subarea lines.

Spare or Redefinable Lines: PUTYPE=4 or PUTYPE=5 is not valid.

For a complete description of this keyword, see PUTYPE on the PU definition statement on page 2-411.

QUIETCT

```
              ┌─QUIETCT=0─────────┐
►►────────────┼───────────────────┼──►◄
              └─QUIETCT=char times─┘
```
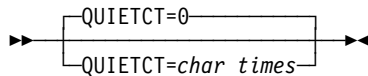
Use: Start-stop lines in emulation mode
Default: 0
Minimum: 0
Maximum: 10

| Specifies, in character times (the time required to transmit 1 character), the interval
| NCP or EP is to wait between the end of a receive operation and the beginning of
the following transmit operation. This pause allows the line to become electrically
quiet following the receive operation.

The default value of 0 is appropriate for most start-stop lines operating at speeds
under 1200 bps. For lines operating at 1200 bps or more, you may need to specify
one or more extra character times to ensure quieting of the line. The recom-
mended value for 1200 bps start-stop lines is 5 (QUIETCT=5).

The pause following a normal receive operation is the value of QUIETCT plus 2.
The pause following the receipt of a negative response to polling is the value of
QUIETCT. Thus, if you specify QUIETCT=5, a normal receive operation is followed
by seven character times, and a negative response to polling is followed by five
character times before the next transmission begins.

REPLYTO

```
                 ┌─seconds─┐
►►───┤ REPLYTO=──┼─────────┼──├──►◄
                 └─NONE────┘
```
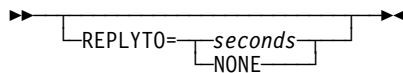
| Use: NCP or EP

Specifies the reply time out value or values for the lines in this group. If NCP or
EP does not receive a response to a poll, selection, or message text before the
specified time expires, a time out error is indicated and the appropriate error
recovery procedure is started.

You can specify this value as an integer or with one decimal place.

When you generate a subarea link, the value of the REPLYTO keyword in the
primary NCP should be longer than the value of the PAUSE keyword in the sec-
ondary NCP.

Reply time outs for start-stop lines are resolved to the nearest half-second, and
reply time outs for BSC lines to the nearest 10th of a second. For example, if you
specify a value of 4.6 for a BSC line group, the reply time out value is 4.6 seconds.
If you specify 4.6 for a start-stop line group, the reply time out value is 4.5 seconds.

Reply time outs for BSC terminals using conversational replies must be long
enough to allow the conversational text to be received.

If any IBM 1050 in the line group includes a paper tape punch, specify a value of at least 23.5 seconds on REPLYTO. If you specify a lesser time, NCP or EP can time out after sending data to the paper tape punch.

If you specify MODE=SECONDARY, do not code REPLYTO.

NONE

If you specify NONE, no time out occurs regardless of the time that elapses between sending to the station and receiving a response.

seconds

Default:	See text
Minimum:	0
Maximum:	60

Specifies the time out interval for the lines in this group. If you code LNCTL=SDLC, the default for REPLYTO is 1. If you code TYPGEN=EP on the BUILD statement and LNCTL=SS on the GROUP statement, the default is 3. If at least one station in the line group is defined as MODULO=128, the default is 3. If you specify MODE=PRI on this GROUP definition statement and there is no LINE or PU definition statement coded beneath it, the default is 1.

For switched SDLC lines, NCP uses the interval specified by XMITDLY on this GROUP definition statement as the reply time out value during initial XID exchange. In other cases, NCP uses the value of REPLYTO.

Attention: If you specify a value greater than 57.5 seconds, NCP might incorrectly report a scanner error.

RETRYTO

```
►►─┬──────────────────┬─►◄
   └─RETRYTO=seconds─┘
```
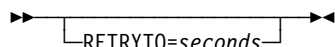
Use:	X.21 interface and network control mode
Default:	See text
Minimum:	0
Maximum:	1632

Specifies, in seconds, how long NCP is to wait after receiving a retry call progress signal (CPS) before retrying a callout. RETRYTO is valid only for X.21 lines (X21NTWK=OLD or X21NTWK=1984 on this GROUP definition statement).

If all the lines in this group are subarea lines (that is, when PUTYPE=4 for each line), the default is RETRYTO=23.5. If any of the lines in this group are not subarea lines, the default is RETRYTO=3.

You can specify this value as an integer or with one decimal place.

RNRLIMT

```
            ┌─RNRLIMT=3─────────────┐
►►──────────┼───────────────────────┼──►◄
            └─RNRLIMT=─┬─minutes─┬───┘
                       └─NONE────┘
```
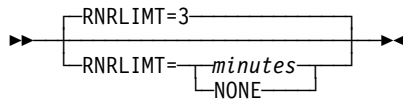
Use: SDLC lines; Token-Ring, frame-relay, and ISDN logical lines
Default: 3
Minimum: 1
Maximum: 90

Specifies, in minutes, how long NCP is to allow a station to refuse to accept data before identifying the station as inoperative. For example, a station is identified as inoperative if it responds with a Receive-Not-Ready (RNR) command to every poll sent by NCP during the specified interval.

RNRLIMT is not valid if you specify LEVEL2 or LNCTL=USER on this GROUP definition statement unless you also specify ECLTYPE or FRELAY.

Token Ring: RNRLIMT is valid only when you specify ECLTYPE=(LOGICAL,PERIPHERAL|SUBAREA).

Frame Relay: RNRLIMT is valid only when you specify FRELAY=LOGICAL.

ISDN: RNRLIMT is valid only when you specify ISDN=(LOGICAL,SUBAREA) or ISDN=(LOGICAL,PERIPHERAL).

TEXTTO

```
►►──┬──────────────────────────┬──►◄
    └─TEXTTO=─┬─seconds─┬───────┘
              └─NONE────┘
```
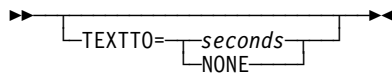
Use: NCP or EP
Default: See text

Specifies the text time out value, in seconds, for the lines in the line group. The maximum is 1632 seconds for lines operating in network control mode and 51.1 seconds for lines operating in emulation mode.

The value defined by TEXTTO is a nominal value. The actual elapsed time can be anything between the nominal value and twice the nominal value. For example, a nominal value of 30 seconds results in an actual interval of 30 to 60 seconds.

Text time outs for start-stop and BSC lines are resolved to the nearest half-second, and text time outs for SDLC links to the nearest 10th of a second. For example, if you specify a value of 20.2 for a start-stop or BSC line group, the text time out value is 20 seconds (nearest half-second). If you specify a value of 4.6 for an SDLC link, the time out is 4.6 seconds.

If you specify TEXTTO=NONE, no time out occurs regardless of the interval that elapses between receipt of successive sets of characters. TEXTTO should be omitted if you specify VIRTUAL=YES or if you specify TERM=MTA on the TERMINAL definition statements of this line group.

For most networks with start-stop and BSC terminals, the default value of 23.5 (or 25.6) seconds is recommended. For SDLC links the recommended value is based

on line speed. Line speeds from 0 to 1199 bps need a time out of about 4 seconds. Line speeds from 1200 to 2000 bps need a time out of about 3 seconds. Line speeds greater than 2000 bps need a 1-second time out.

If you define a subarea link (PUTYPE=4), do not specify TEXTTO=NONE. If you define an SDLC link in secondary mode (MODE=SECONDARY), code TEXTTO=NONE.

Attention: Diagnostic programs for the IBM 2845 and 2848 control units attached to lines within the group will fail if you define a value of more than 2 seconds for TEXTTO or omit the keyword.

Table 2-36 shows the effect on TEXTTO when you code certain keywords on the GROUP and TERMINAL definition statements.

Table 2-36. NDF Values for TEXTTO on the GROUP Definition Statement

When You Code This	TEXTTO Defaults To
GROUP: LNCTL=SDLC	3.0
GROUP: NCST=IP	3.0
GROUP: ETHERNET	3.0
GROUP: FRELAY	3.0
GROUP: LNCTL=SS and TYPE=EP	25.6
TERMINAL: TERM=2741	NONE

TIMER
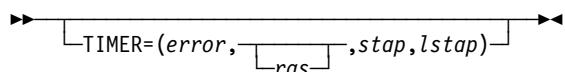
```
►►──┬────────────────────────────────────────────┬──►◄
    └─TIMER=(error,──┬─────┬──,stap,lstap)─┘
                     └─ras─┘
```

Use: User-written line control and NTRI physical lines
Default: None

Specifies the entry points of user-written or NTRI timer service routines. Code TIMER if you specify LEVEL2 and LEVEL3. If you do not include level 2 and level 3 user-written or NTRI code, or if you specify CHANLNK=YES on the GROUP definition statement, omit TIMER. All four suboperands are required.

error
    Specifies the entry point of the user-written or NTRI timer error service routine.

ras
    Specifies the entry point of the user-written timer RAS service routine used to make periodic checks for certain error conditions. Although *ras* is optional, include the comma whether or not you code *ras*.

stap
    Specifies the entry point of the user-written or NTRI timer shoulder-tap service routine that receives control when a user-initiated timer reaches 0.

lstap
    Specifies the entry point of the user-written or NTRI lagging shoulder-tap timer routine.

NTRI physical lines: If you code TIMER in the definition of a NTRI physical line group, specify TIMER=(ECLNART1,,ECLNART2, ECLNART3). Otherwise, NDF will add this value for you.

Note: TIMER is ignored for Token-Ring logical line groups (ECLTYPE=LOGICAL) and frame-relay logical line groups (FRELAY=LOGICAL).

TTDCNT
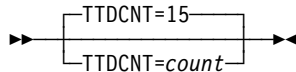
```
              ┌─TTDCNT=15───┐
►►────────────┤             ├──────►◄
              └─TTDCNT=count─┘
```

Use: BSC lines in network control mode
Default: 15
Minimum: 1
Maximum: 255

Specifies how many times NCP is to accept the BSC temporary text-delay (TTD) sequence from a device before aborting the operation. The TTD sequence notifies NCP that the device is temporarily unable to send the next block of data. Code TTDCNT=255 to accept the sequence an unlimited number of times.

For most lines, the default value of 15 TTD transmissions is recommended.

TYPE

```
►►──────┬──────────────────┬──────►◄
        └─TYPE=──┬─NCP─┬────┘
                 ├─PEP─┤
                 └─EP──┘
```
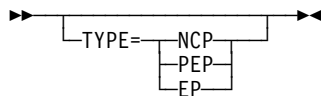
| Use: NCP or EP
Default: See text

Specifies whether the lines in this group operate in network control mode, emulation mode, or both.

- Specify TYPE=NCP if all the lines in the group always operate in network control mode. (A group of SDLC lines can operate only in network control mode.) When you code TYPE=NCP, keywords that apply only to emulation mode are ignored on this GROUP definition statement.

- Specify TYPE=PEP if some lines in the group operate in network control mode and others in emulation mode, or if any lines can operate in either mode.

- Specify TYPE=EP if all the lines in the group always operate in emulation mode. When you code TYPE=EP, keywords that apply only to network control mode are ignored on this GROUP definition statement. For EP standalone users, TYPE=EP is the only valid value.

VTAM does not provide any default processing for TYPE, so if the NCP generation definition is used as input to VTAM, specify a value for TYPE.

**Spare or Redefinable Lines:** Only TYPE=NCP is valid.

Table 2-37 on page 2-137 shows the effect on TYPE when you code certain keywords on the BUILD and GROUP definition statements.

Table   2-37. NDF Values for TYPE on the GROUP Definition Statement

When You Code This	The Effect on TYPE Is
BUILD: TYPGEN=NCP or NCP-R	The default is TYPE=NCP. The only valid value is TYPE=NCP.
BUILD: TYPGEN=PEP and GROUP: LNCTL=SS or BSC	The default is TYPE=PEP.
BUILD: TYPGEN=EP	The default is TYPE=EP. The only valid value is TYPE=EP.
GROUP: LNCTL=SDLC	The default is TYPE=NCP. The only valid value is TYPE=NCP.
GROUP: ECLTYPE	NDF adds TYPE=NCP. The only valid value is TYPE=NCP.
GROUP: NCST=IP	The default is TYPE=NCP. The only valid value is TYPE=NCP.
GROUP: ETHERNET=PHYSICAL	The default is TYPE=NCP. The only valid value is TYPE=NCP.
GROUP: FRELAY	NDF adds TYPE=NCP. The only valid value is TYPE=NCP.
GROUP: ISDN	NDF adds TYPE=NCP. The only valid value is TYPE=NCP.

USERID

Suboperands:

Use: NTRI or user-written code

Specifies an identification symbol given to user-defined or NTRI control blocks associated with this line group. The value also can specify the block dump-table entry label and the type of RECMS processing to be performed for this line group.

id

Use: Required if you code USERID

Specifies the identification symbol given to user-defined or NTRI control blocks associated with this line group. *id* is a symbol consisting of 1 to 7 alphanumeric characters. This identification is placed in the group control block for this line group.

entry

Default: None

Specifies the entry label of the block dump table to be used by the dump formatter to format user-defined or NTRI control blocks. If you omit *entry*, the dump formatter does not format the control blocks.

NORECMS

Default: None

Specifies that NCP is not to perform RECMS processing. Therefore, you must supply a RECMS routine if you wish to perform RECMS processing. USRECMS is no longer valid as the third suboperand. If you specify USRECMS, NDF ignores it.

traceid

Default: U

Specifies the unique trace identification symbol for user-written or NTRI code. This identification is placed in the group control block for this line group. *traceid* is a single alphanumeric character.

entry2

Default: None

Specifies the name of the user-written or NTRI NMVT BUILD routine to be invoked when a station or link has been identified as inoperable.

**3746 Model 900 Token Ring:** The USERID keyword is ignored.

Table 2-38 shows the effect on USERID when you code certain keywords on the GROUP definition statement.

Table 2-38. NDF Values for USERID on the GROUP Definition Statement

When You Code This on GROUP	The Effect on USERID Is
ECLTYPE=PHYSICAL (NTRI only)	NDF adds USERID=(5668854,ECLRBDT, NORECMS,*traceid*,ECLNMVT).
ECLTYPE=(LOGICAL,PERIPHERAL|SUBAREA) (NTRI only)	NDF adds USERID=(5668854,ECLVBDT, NORECMS,traceid,ECLNMVT).
ECLTYPE=(LOGICAL,FRELAY)	NDF adds USERID=(5668854,ECLFTDT, NORECMS,traceid,ECLNMVT).

VIROWNER

►►──────────────────────►◄
　　└─VIROWNER=*name*─┘

Specifies the name of the user-written generation load module that is responsible for processing the resources defined by this GROUP definition statement. *name* must match the name of one of the user-written generation load modules specified by USERGEN on the OPTIONS definition statement. The start record of this load module's statement and keyword vector table (SKVT) must indicate that the load module is valid for VIROWNER.

VIROWNER is valid only if you code VIRTUAL=YES on this GROUP definition statement.

VIRTUAL

```
                   ┌─VIRTUAL=NO─┐
►►─────────────────┼────────────┼────────────►◄
                   └─VIRTUAL=──┬─YES─┬─┘
                              └─NO──┘
```
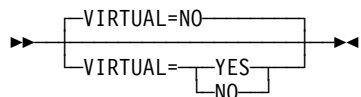
Default: NO

Specifies whether NCP is to support programmed resources in this group.
PUTYPE=4 is not valid if you code VIRTUAL on the GROUP definition statement.
VIRTUAL is valid only when LNCTL=SDLC.

Spare or Redefinable Lines: Only VIRTUAL=NO is valid.

Table 2-39 shows the effect on VIRTUAL when you code certain keywords on the
GROUP definition statement.

Table   2-39. NDF Values for VIRTUAL on the GROUP Definition Statement

When You Code This on GROUP	The Effect on VIRTUAL Is
ECLTYPE	The only valid value is VIRTUAL=NO.
NCST=IP	VIRTUAL=YES is required.
ETHERNET=PHYSICAL	The default is VIRTUAL=NO. The only valid value is VIRTUAL=NO.
FRELAY	The default is VIRTUAL=NO. The only valid value is VIRTUAL=NO.
ISDN	The default is VIRTUAL=NO. The only valid value is VIRTUAL=NO.

WACKCNT
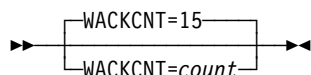
```
                   ┌─WACKCNT=15──────┐
►►─────────────────┼─────────────────┼────►◄
                   └─WACKCNT=count──┘
```

Use: BSC lines in network control mode
Default: 15
Minimum: 1
Maximum: 255

Specifies how many times NCP is to accept the wait-before-transmit positive
acknowledgment (WACK) from a BSC station before NCP aborts the operation.
The WACK sequence notifies NCP that the device is temporarily not ready to
receive. Code WACKCNT=255 to accept the sequence an unlimited number of
times.

For most BSC lines, the default value of 15 WACK transmissions is recommended.

WAKDLAY

```
                   ┌─WAKDLAY=2.2─┐
►►─────────────────┴─────────────┴──────►◄
        └─WAKDLAY=─┬─seconds─┬─┘
                   └─NONE────┘
```
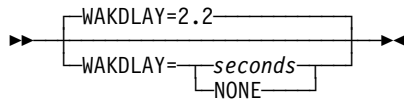
Use: BSC lines in network control mode
Default: 2.2
Maximum: 1632

Specifies, in seconds, how long NCP can delay before responding to message text received from a device on any line in this group. If NCP is unable to respond with a positive acknowledgment before this interval elapses, NCP sends a WACK sequence instead.

If you code WAKDLAY=NONE, NCP does not send a WACK sequence when unable to send a positive acknowledgment.

You can specify this value as an integer or with one decimal place.

For most BSC lines, the default interval of 2.2 seconds is recommended.

WTTYEOB
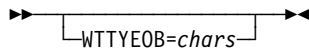
```
►►─────────────────────►◄
    └─WTTYEOB=chars─┘
```

Use: WTTY lines in network control mode
Default: None

Specifies the hexadecimal representation of the EOB sequence required for WTTY teleprinters connected to lines in this group operating in network control mode. Define up to 8 hexadecimal characters (4 EBCDIC characters). You can include any valid character in the sequence, but include the FIGS and LTRS characters only once. If the terminal is equipped to send who-are-you (WRU), do not use the letter *D* in the sequence. See the "Line Character Codes" section of *NCP and EP Reference Summary and Data Areas*, Volume 2, for the WTTY transmission codes.

WTTYEOT
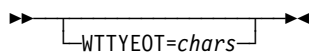
```
►►─────────────────────►◄
    └─WTTYEOT=chars─┘
```

Use: WTTY lines in network control mode
Default: None

Specifies the hexadecimal representation of the end-of-transmission (EOT) sequence required for WTTY teleprinters connected to lines in this group operating in network control mode. You can include any valid character in the sequence, but include the FIGS and LTRS characters only once. If the terminal is equipped to send who-are-you (WRU), do not use the letter *D* in the sequence.

WTTYEOT is required for all WTTYs. See the "Line Character Codes" section of *NCP and EP Reference Summary and Data Areas*, Volume 2, for the WTTY trans-mission codes.

XIO
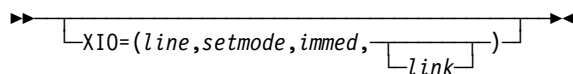
```
►►──┬────────────────────────────────────┬──►◄
    └─XIO=(line,setmode,immed,──┬──────┬──)─┘
                                └─link─┘
```

Use: User-written line control, NTRI, and frame relay

Specifies the entry points of user-written, NCP, or NTRI execute-I/O service rou-
tines. XIO is required if you specify LEVEL2 and LEVEL3. If you do not include
level 2 and level 3 user-written code, omit XIO.

line
Specifies the entry point of the execute-I/O line service routine. Required if you
code XIO.

setmode
Specifies the entry point of the execute-I/O set-mode service routine. Required
if you code XIO

immed
Specifies the entry point of the execute-I/O immediate service routine.
Required if you code XIO

link
Specifies the entry point of the execute-I/O link service routine. This entry point
is required if either user-written code or NCP level 5 code issues an XIO LINK
command. If no XIO LINK commands are issued, you can omit this parameter.

3746 Model 900 Token Ring: The XIO keyword is ignored.

3746 Model 900 Frame Relay: The XIO keyword is ignored.

Table 2-40 shows the effect on XIO when you code certain keywords on the
GROUP definition statement.

Table 2-40. NDF Values for XIO on the GROUP Definition Statement

When You Code This on GROUP	The Effect on XIO Is
ECLTYPE=PHYSICAL (NTRI only)	NDF adds XIO=(ECLNARXL,ECLNARXS, ECLNARXI,ECLNARXK). The only valid value is XIO=(ECLNARXL, ECLNARXS,ECLNARXI,ECLNARXK).
ECLTYPE=LOGICAL (NTRI only)	NDF adds XIO=(ECLNAVXL,ECLNAVXS, ECLNAVXI,ECLNAVXK). The only valid value is XIO=(ECLNAVXL, ECLNAVXS,ECLNAVXI,ECLNAVXK).
FRELAY=LOGICAL (3745 only)	NDF adds XIO=(ECLNAVXL,ECLNAVXS, ECLNAVXI,ECLNAVXK). The only valid value is XIO=(ECLNAVXL, ECLNAVXS,ECLNAVXI,ECLNAVXK).

XMITDLY

```
►►─┬─────────────────────────┬─►◄
   └─XMITDLY─┬─seconds─┬──────┘
            └─NONE────┘
```
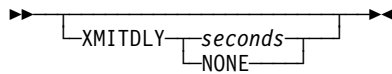
Use: SDLC lines in network control mode
Default: See text
Minimum: See text
Maximum: 60

Specifies, in seconds, how long NCP is to delay its initial transmission after answering an incoming call. You can specify this value as an integer or with one decimal place. If you do not want NCP to delay its initial transmission, code XMITDLY=NONE.

The default is either 2.2 seconds or the value of REPLYTO on this GROUP definition statement, whichever is greater. The minimum is the value of REPLYTO on this GROUP definition statement. If you do not code REPLYTO, the minimum is 1.

XMITDLY is valid only if you specify DIAL=YES and LNCTL=SDLC on this GROUP definition statement. The value of XMITDLY is used as the reply time-out value during initial XID exchange (see REPLYTO on the GROUP definition statement on page 2-132 for more information).

3746 Model 900 Frame Relay: The XMITDLY keyword is ignored.

Table 2-41 shows the effect on XMITDLY when you code certain keywords on the GROUP definition statement.

Table 2-41. NDF Values for XMITDLY on the GROUP Definition Statement

When You Code This on GROUP	The Effect on XMITDLY Is
ECLTYPE=(LOGICAL,PERIPHERAL|FRELAY) (NTRI only) [1]	NDF adds XMITDLY=NONE. The only valid value is XMITDLY=NONE.
FRELAY=(LOGICAL,PERIPHERAL) (3745 only)	NDF adds XMITDLY=NONE. The only valid value is XMITDLY=NONE.

[1] The XMITDLY keyword is ignored for Token-Ring resources when ADAPTER=TIC3.

X21NTWK

```
      ┌─X21NTWK=NO──────────┐
►►─┬──┴─────────────────────┴──┬─►◄
   └─X21NTWK=─┬─NO───┬──────────┘
             ├─OLD──┤
             └─1984─┘
```
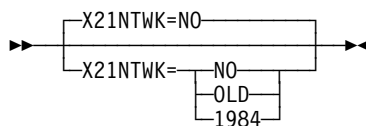
Use: X.21 lines
Default: NO

Specifies whether this group is attached to an X.21 network and specifies the level of the X.21 network. X21NTWK=NO specifies that this group is not attached to an X.21 network. X21NTWK=OLD specifies that the X.21 network is at pre-1984 level. X21NTWK=1984 specifies that the X.21 network is at the 1984 level.

If you code X21NTWK, the following restrictions apply to the LINE definition statement:

- Do not code the AUTO and REDIAL keywords.

- Specify FULL as the second suboperand of the ADDRESS keyword if DIAL=YES on this GROUP definition statement. If DIAL=NO or if DIAL is not coded, you can specify HALF or FULL on the ADDRESS keyword.

- Specify X21SW=YES if DIAL=YES.

- Specify ANSTONE=NO, CLOCKNG=EXT, NEWSYNC=NO, and RING=NO.

X21SW

```
               ┌─X21SW=NO─────┐
►►─────────────┼──────────────┼──►◄
               └─X21SW=─┬─NO─┬─┘
                        └─YES─┘
```
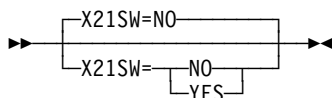
Use: X.21 switched lines
Default: NO

Specifies whether the lines in this group are X.21 switched lines. VTAM uses this information.

X21SW is required if the X21NTWK keyword is coded and DIAL=YES.

GWNAU Definition Statement

Usage: SNA network interconnection. One for each predefined resource that requires a cross-network address representation, and one for each network that requires a pool of reusable cross-network addresses.

Position: For logical units and SSCPs in non-native networks that require addresses in the native network, the GWNAU definition statement must follow the NCPNAU or SYSCNTRL definition statement. For logical units and SSCPs in any network (native or non-native) that require addresses in a non-native network, the GWNAU definition statement must follow the NETWORK definition statement for the non-native network.

NDF requires that the GWNAU definition statement that generates the pool of network addressable units (NAUs) must be the last definition statement in the sequence of GWNAU or NCPNAU definition statements within a network. If you specify a nonzero value for DYNPOOL on the BUILD definition statement, you do not need to code a GWNAU definition statement following each NETWORK or BUILD definition statement.

Keywords: Table 2-42 shows the keywords described under this definition statement.

*Table 2-42. Keywords on the GWNAU Definition Statement*

Keyword	Page	Required	Sift	Default	Tuning*
ELEMENT	2-145				
NAME	2-145				
NETID	2-146				
NUMADDR	2-146				
NUMSESS	2-146			1	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions
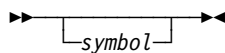
symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

ELEMENT

```
►►──────────────────────►◄
      └─ELEMENT=address─┘
```
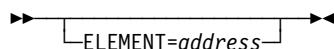
Use: Predefined cross-network resource addresses
Default: None
Maximum: 65534

Specifies the specific element address to be assigned to this resource when sequential addressing is not desired. Code ELEMENT only when a specific element address is necessary.

ELEMENT is valid only if you code NAME and NETID on this GWNAU definition statement.

GWNAU definition statements on which ELEMENT is coded must precede all other GWNAU definition statements for each network and must be in ascending order by *address*.

The values of the ELEMENT keywords need not be consecutive; for example, you might code a sequence of GWNAU definition statements with ELEMENT values of 2, 5, 6, 12, and so on. However, each address you skip wastes 8 bytes of storage and addresses. Resources following the last GWNAU definition statement on which ELEMENT is coded are assigned sequential element addresses starting at the highest address you specify plus 1.

VTAM: If this GWNAU definition statement represents a logical unit session, the value coded for ELEMENT must be different from addresses coded on the GWPATH definition statement or cross-domain resource manager (CDRM) definition in VTAM. If this GWNAU definition statement represents an SSCP session, the value coded for ELEMENT must match the address coded on the GWPATH definition statement or CDRM definition in VTAM.

NPM: Assign consecutive element numbers to obtain reliable performance information.

TCAM: It is recommended that you do not code ELEMENT because TCAM requires consecutive element addresses. If you code ELEMENT, its value on the first GWNAU definition statement must be the first available element address, and each subsequent value must increase by one.

NTO: Because the NTO uses element number 1, do not code ELEMENT=1.

NAME

```
►►────────────────────►◄
      └─NAME=──name──┘
```
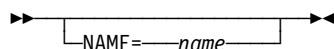
Use: Predefined cross-network resource addresses
Default: None

Specifies a name to be associated with the address being defined. *name* is the name of the SSCP or logical unit that the address represents as it is known in its owning network. You can define more than one GWNAU with the same name.

NAME is valid only if you code NETID on this GWNAU definition statement. If you omit NAME, code NUMADDR.

NETID

```
►►──┬──────────────┬──►◄
    └─NETID=name─┘
```
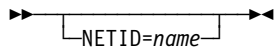
Use: Predefined cross-network resource addresses
Default: None

Specifies a network identifier to be associated with a resource address. *name* is the name of the network in which the SSCP or logical unit that the address represents resides. *name* can be 1 to 8 characters long.

NETID is valid only if you code NAME on this GWNAU definition statement. If you omit NETID, specify NUMADDR.

NUMADDR

```
►►──┬────────────────────┬──►◄
    └─NUMADDR=number─┘
```
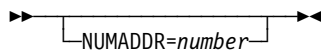
Use: A pool of reusable cross-network addresses
Default: None
Minimum: 1
Maximum: See text

Specifies the number of addresses to be reserved for cross-network resource address assignment. NUMADDR is not valid if you code ELEMENT, NAME, NETID, or NUMSESS on this GWNAU definition statement.

You can code NUMADDR on only one GWNAU definition statement in each network. This GWNAU definition statement must be the last GWNAU definition statement in the network.

The maximum value of NUMADDR depends on the amount of NCP storage and the element addresses available for subareas in each attaching network.

Note: You can define part of the NCP buffer pool to support additional cross-network addresses when the defined capacity for cross-network addresses has been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

NUMSESS

```
      ┌─NUMSESS=1───────┐
►►──┼─────────────────────┼──►◄
      └─NUMSESS=number─┘
```
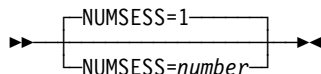
Use: Predefined cross-network resource addresses
Default: 1
Minimum: 0
Maximum: 255

Specifies the number of half-session control blocks (HSCBs) to be permanently assigned to the address being defined.

The total of all NUMSESS values for the native network cannot exceed 65535, and the total of the NUMSESS values for all non-native networks cannot exceed 65535. NUMSESS is valid only if you define NAME and NETID on this GWNAU definition statement.

NUMSESS and SESSLIM interact on the BUILD or NETWORK definition statements if the GWNAU definition statement defines a permanent assignment for a specific SSCP or logical unit, and the BUILD or NETWORK definition statement defines a pool of control blocks to be dynamically assigned. In this case, if the value of NUMSESS is less than the value of SESSLIM, the gateway NCP allocates additional control blocks from the pool to the SSCP or logical unit until the value of SESSLIM is reached or until the number of control blocks taken from the pool reaches the value of NETLIM.

If you specify a value greater than or equal to the value of SESSLIM, the gateway NCP allocates the requested number of control blocks but does not allocate additional control blocks from the pool, even if needed.

HOST Definition Statement

Usage: NCPs in channel-attached controllers. One for each host that activates NCP. This definition statement is for VTAM only. It is not required in the generation definition you use to generate NCP.

Position: Must precede the first GROUP definition statement.

Keywords: See the *VTAM Resource Definition Reference* for the description of this definition statement and its keywords.

IDLIST Definition Statement

Usage: BSC or TWX devices on switched telecommunication links. One for each list of identifier sequences checked by NCP. The IDLIST definition statement is used only in an NCP that communicates with TCAM. IDLIST is not valid for VTAM.

Position: Must precede the first GROUP definition statement.

Keywords: Table 2-43 shows the keywords described under this definition statement.

Table 2-43. Keywords on the IDLIST Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
IDSEQ	2-149	Yes			
MAXLEN	2-150				
NOMATCH	2-151			PASS	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions
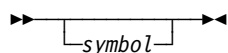
symbol

Default: None

Provides a name for the identifier list and is required except as indicated in the description of IDSEQ. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

IDSEQ
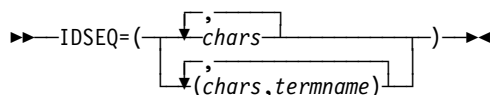
Use: Required when you code the IDLIST definition statement

Specifies the identification sequence for each device that can call or be called by the controller. IDSEQ also can specify the name of the TERMINAL definition statement for the device from which the sequence is expected.

chars

Default: None

Specifies an identification sequence NCP is to recognize as valid. Code *chars* as the hexadecimal representation of the EBCDIC characters to be recognized. You can specify a maximum of 20 EBCDIC characters in one sequence.

Attention: Any EOT, ENQ, or ACK characters sent by a TWX terminal as part of its identifier sequence are deleted by NCP as it receives the sequence into a buffer. Do not include any of these characters in the identifier sequence you specify for IDSEQ or NCP will fail to recognize the received sequence.

termname

Default: None

Specifies the name of the TERMINAL definition statement for the device to be associated with the identifier characters. Do not specify the name of a TER-MINAL definition statement on which you code CTERM=YES.

If you specify a terminal name for any identification sequence, you must specify a terminal name for *all* sequences.

If you omit *termname*, NCP recognizes *chars* as a valid sequence but does not recognize it as the sequence for a specific device.

You can code a maximum of 255 characters in IDSEQ, including the beginning and ending parentheses and all commas. This limit applies regardless of how many identifier sequences and terminal names you specify. If you need more than 255 characters, code additional IDLIST definition statements (omitting the symbol of each) directly following the first IDLIST definition statement. (Only the first IDLIST definition statement can include MAXLEN and NOMATCH.)

MAXLEN

```
►►──┬──────────────┬──►◄
    └─MAXLEN=bytes─┘
```
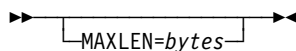
Default: None
Maximum: 256
Minimum: 8

Specifies the maximum size of the list of identifier sequences specified by IDSEQ, in bytes (*not* the number of entries). This value includes the total number of bytes in all entries, plus the control field that precedes the list. Specify MAXLEN only if the access method uses the dynamic control facility either to add entries to the list or to increase the size of one or more existing entries in the list.

Calculate the value for MAXLEN as follows:

1. Add 4 bytes for the list header.

2. For each entry in the list, add the number of bytes required to contain the iden-tifier sequence, plus 2 bytes if you omit *termname* in IDSEQ or 4 bytes if you specify *termname* in IDSEQ.

3. Round the size of each entry to the next higher fullword.

NOMATCH

```
           ┌─NOMATCH=PASS───────┐
►►─────────┼────────────────────┼─►◄
           └─NOMATCH=──┬─PASS─┬─┘
                       └─STOP─┘
```
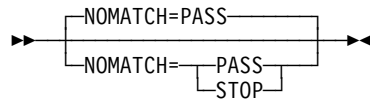
Default: PASS

Specifies the action NCP is to perform if it does not receive an identifier sequence from a calling device, or if it receives an identifier sequence it does not recognize as valid (a sequence not defined by this IDLIST definition statement).

If NCP is to send any unrecognized identifier sequences to the host processor, code NOMATCH=PASS. If NCP receives no identifier sequence, this fact is indicated by the response returned to the access method.

If NCP is not to send unrecognized sequences to the host processor, code NOMATCH=STOP. In this case, if NCP receives an identifier sequence it does not recognize, it breaks the line connection.

IPHOST Definition Statement

Usage: For Internet Protocol. Defines the hardware address of an internet host on a Token-Ring or Ethernet-type LAN as a permanent Address Resolution Protocol (ARP) table entry. Use a permanent ARP table entry either to define a device that does not support ARP packets or to avoid sending ARP requests. When you use IPHOST to define the address of an internet host attached to a Token Ring, the host must reside on the local Token-Ring segment and cannot be moved from that Token Ring. You can code more than one IPHOST definition statement for each physical connection to a Token-Ring or Ethernet-type LAN.

IPHOST is not valid for internet connections to a frame-relay network, for IP channel connections, or for internal NCP to Model 900 point-to-point IP connections.

Position: All IPHOST definition statements that define devices on a particular Token-Ring or Ethernet-type LAN must immediately follow the LINE and PU definition statements that define the connection to that LAN.

Keywords: Table 2-44 shows the keywords described under this definition statement.

Table 2-44. Keywords on the IPHOST Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
HDWADDR	2-153	Yes			
IPADDR	2-153	Yes			
LANTYPE	2-154				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►──┬──────────┬──►◄
     └─symbol──┘
```
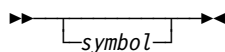
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

HDWADDR

►►——HDWADDR=*address*——►◄

Use: Required when you code the IPHOST definition statement
Default: None

Specifies the hardware address of the internet host device defined by this IPHOST definition statement. This hardware address for a device attached to a Token-Ring or Ethernet-type LAN is associated with the internet address specified on the IPADDR keyword.

address must be specified as a character string of 12 hexadecimal digits. Use the format HDWADDR=*abbbbbbbbbbb*, where *a* is a value between 0 and 7, and *b* is a value between 0 and F. The bits in this address have the following meaning:

Bit 0   0 indicates an individual address.
Bit 1   0 indicates a universally administered address.
             1 indicates a locally administered address.
Bits 2–47   Specify the hardware address of the device.

Because NCP does not support group addresses as destination medium access control (MAC) addresses, the group address indicator (bit 0) must have a value of zero. The valid range for HDWADDR is X'000000000001'–X'7FFFFFFFFFFF'; hardware addresses in the range X'800000000000'–X'FFFFFFFFFFFF' are not valid.

To determine the hardware address for a particular workstation, see the workstation documentation.

IPADDR

►►——IPADDR=*address*——►◄

Use: Required when you code the IPHOST definition statement
Default: None

Specifies the internet address for the internet host defined by this IPHOST definition statement. This value is associated with the hardware address specified on the HDWADDR keyword.

address is a 4-byte internet address that can be specified in either dotted decimal or hexadecimal format.

To code the address as a dotted decimal number, use the format IPADDR=*n0.n1.n2.n3* where *n0–n3* are numbers between 0 and 255, for example, IPADDR=128.255.16.1.

To code the address as a hexadecimal number, use the format IPADDR=*hhhhhhhh*. For example, code the address 128.255.16.1 as IPADDR=80FF1001. See the *NCP, SSP, and EP Resource Definition Guide* for more information about internet addresses.

LANTYPE

```
►►─┬───────────────────────────┬─►◄
   └─LANTYPE=─┬─802.3─┬─────────┘
             └─ENETV2─┘
```
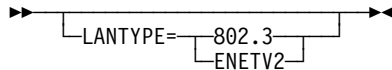
Use: Ethernet-type LAN hosts
Default: See text

Specifies the type of frame format used by the internet host attached to an Ethernet-type LAN.

802.3
Supports the IEEE 802.3 standard frame format only.

ENETV2
Supports the Ethernet Version 2 frame format only.

Required if LANTYPE=DYNAMIC is specified on the PU definition statement. Optional if either LANTYPE=802.3 or LANTYPE=ENETV2 is specified on the PU definition statement. If either LANTYPE=802.3 or LANTYPE=ENETV2 is specified on the PU definition statement, that value is the default for LANTYPE on the IPHOST definition statement. LANTYPE is not valid when you are defining a hardware address for a Token-Ring-attached internet host.

IPLOCAL Definition Statement

Usage: For Internet Protocol.

Note: For frame-relay IP interfaces and internal NCP to Model 900 point-to-point IP connections, you can specify keywords on the PU definition statement that usually are specified on an IPLOCAL definition statement. The same is true for channel interfaces, unless the channel is used for direct communication with NCPROUTE.

Defines the characteristics of an internet interface that passes datagrams between an NCST session, a NTRI or Ethernet-type LAN interface, a frame-relay interface, a channel interface, an internal NCP to Model 900 point-to-point IP connection, and the NCP internet router.

When you code an IPLOCAL definition statement that defines an interface to a locally attached NTRI or Ethernet-type LAN, NCP implicitly defines a route that can be used to forward traffic to destinations on the locally attached LAN or subnetwork of the LAN. For NCST, frame-relay connections, internal NCP to Model 900 point-to-point IP connections, and channel connections, NCP implicitly defines a route to the internet host.

Note: A NTRI or Ethernet-type LAN interface or a frame-relay interface defined by the IPLOCAL definition statement can support internet broadcasts; however, NCST interfaces, channel interfaces, and internal NCP to Model 900 point-to-point IP connections are considered point-to-point connections and do not support internet broadcasts.

Position: Definition statements used to define internet routing information to NCP must appear after the last line group and precede the GENEND definition statement in the following order.

1. One IPOWNER definition statement
2. All IPLOCAL definition statements
3. All IPROUTE definition statements

Keywords: Table 2-45 shows the keywords described under this definition statement.

Table 2-45. Keywords on the IPLOCAL Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
INTFACE	2-156	Yes			
LADDR	2-156	Yes			
METRIC	2-157				
P2PDEST	2-157				
PROTOCOL	2-158			RIP	
SNETMASK	2-158				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►─────────────────►◄
    └─symbol─┘
```
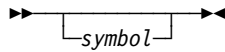
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

INTFACE

```
►►──INTFACE=name──►◄
```

Use: Required when you code the IPLOCAL definition statement
Default: None

Specifies the name of the internet interface for which the local internet address is being defined. The name must be 1 to 8 characters long and must be the same as the value specified by one of the following:

- INTFACE keyword on the LINE definition statement defining a NTRI or Ethernet-type LAN attachment
- INTFACE keyword on the LU definition statement defining an NCST session interface
- INTFACE keyword on the PU definition statement defining a frame-relay inter-face, channel interface, or internal NCP to Model 900 point-to-point IP con-nection.

LADDR
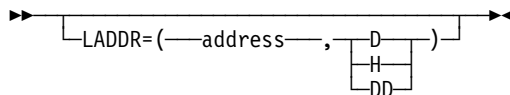
```
►►──────────────────────────────►◄
    └─LADDR=(──address──,──┬─D──┬──)─┘
                          ├─H──┤
                          └─DD─┘
```

Use: Required when you code the IPLOCAL definition statement
Default: None

Specifies the internet address of the associated interface.

address is a 4-byte internet address that can be specified in either dotted decimal or hexadecimal format. The value you specify for LADDR cannot be specified on any other IPLOCAL definition statement. The address specified must be in the same network or subnetwork as the locally attached network or subnetwork. The network and subnetwork portions of the LADDR address must match the address of the associated network or subnetwork.

To code the address as a decimal number, use the format
LADDR=(*dddddddddd*,D) (D means decimal representation). For example, code
the address 128.255.16.64 as LADDR=(2164199488,D).

Note: The value 2147483648 (2^{31}) is not valid.

To code the address as a hexadecimal number, use the format LADDR=*hhhhhhhh*
or LADDR=(*hhhhhhhh*,H) (H means hexadecimal representation). For example,
code the address 128.255.16.64 as LADDR=80FF1040 or LADDR=(80FF1040,H).

To code the address as a dotted decimal number, use the format
LADDR=*n0.n1.n2.n3* or LADDR=(*n0.n1.n2.n3*,DD) (DD means dotted decimal rep-
resentation) where *n0–n3* are numbers between 0 and 255. For example,
LADDR=128.255.16.64 or LADDR=(128.255.16.64,DD).

The address specified by LADDR is associated with this NCP and can be a desti-
nation internet address for certain Transmission Control Protocol/Internet Protocol
(TCP/IP) functions. For more information, see "IP Datagram Fragmentation and
Reassembly" in Chapter 7 of the *NCP and EP Reference*.

See the *NCP, SSP, and EP Resource Definition Guide* for more information about
internet addresses.

METRIC

►►──METRIC=*value*──►◄

Use: Internet Protocol
Default: See Text
Minimum: 1
Maximum: 15

Specifies a value used by NCPROUTE to evaluate new route information received
in Routing Information Protocol (RIP) packets over the interface defined by the
IPLOCAL definition statement.

If the values coded for INTFACE and P2PDEST on the IPLOCAL definition state-
ment match the values coded for INTFACE and HOSTADDR on the IPOWNER
definition statement, and an IPROUTE definition statement is coded with the same
values for INTFACE and DESTADDR, respectively, NCP will use the METRIC
value coded on the IPROUTE definition statement when the route is built.

For frame-relay IP connections, channel IP connections, and internal NCP to Model
900 point-to-point IP connections, METRIC is an optional keyword. If you do not
specify a METRIC value, a value of 1 is used.

P2PDEST

►►──────────────────────►◄
    └─P2PDEST=*host_address*─┘

Use: Required for NCST IP connections, channel IP connections, and
           internal NCP to Model 900 point-to-point IP connections.
Default: None

IPLOCAL

Specifies the internet host address of the destination NCST logical unit or internet host in a point-to-point connection.

For an NCST connection, if the INTFACE value on an IPLOCAL definition statement matches the INTFACE value on the IPOWNER definition statement, the P2PDEST address should also match the HOSTADDR value on the IPOWNER definition statement.[3] Be sure that the P2PDEST address and the LADDR address on an IPLOCAL definition statement belong to the same network or subnetwork.

Note: P2PDEST is not valid for NTRI and Ethernet-type LAN connections.

For frame-relay internet connections, the P2PDEST keyword is optional on the IPLOCAL definition statement. When P2PDEST is not specified, Inverse ARP processing is initiated to resolve the destination internet address.

PROTOCOL

```
             ┌─PROTOCOL=RIP─────────┐
►►───────────┼──────────────────────┼───►◄
             └─PROTOCOL=─┬─RIP─┬─────┘
                         └─NONE─┘
```
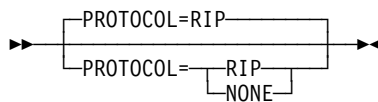
Use: Internet Protocol
Default: RIP

Specifies that RIP can manage information about the router interface specified on the IPLOCAL definition statement. NCP uses the NCPROUTE program and RIP to transmit information about this interface to other routers. Code PROTOCOL=NONE when you define interfaces that use other protocols.

Note: RIP Version 2 is supported starting with NCP V7R6. The version of RIP to be used is configured in the NCPROUTE program provided by IBM TCP/IP.

SNETMASK
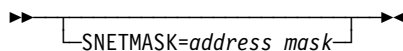
```
►►───┬──────────────────────────┬───►◄
     └─SNETMASK=address_mask─────┘
```

Use: Internet Protocol subnetworks
Default: See text

Specifies a unique subnetwork mask. The subnetwork mask determines which portion of an internet address, defined by the LADDR keyword, identifies the subnetwork associated with this internet interface. The SNETMASK value must contain the network portion of the internet address defined by the LADDR keyword.

When VSUBNETS=NO on the BUILD statement:
> You can define a subnetwork mask for each network by coding SNETMASK on the first IPLOCAL (or PU) definition statement that defines an interface to that network. If you omit SNETMASK on subsequent IPLOCAL (or PU) definition statements for that network, the SNETMASK value coded on the first IPLOCAL (or PU) definition statement with a LADDR value for that network is the default.

---

[3] This restriction does not apply in the absence of SNALINK, where the remote NCP definition must use an indirect path to NCPROUTE via an NCST connection to a local NCP that has an IP channel connection to the NCPROUTE host.

When VSUBNETS=YES on the BUILD statement:

> You can define one or more subnet masks for each network by coding SNETMASK on one or more of the IPLOCAL (or PU) statements that define interfaces to that network. If you omit SNETMASK on any of the IPLOCAL (or PU) definition statements for that network, a default value will **not** be supplied, and that interface will not have subnetting.

Internal NCP to Model 900 Point-to-Point IP Connections: The 3746 Model 900 Controller Configuration and Management (CCM) definitions will also include the IP subnetwork mask for the 3746 Model 900 side of the internal NCP to Model 900 point-to-point IP connection. This CCM-defined address must match the corresponding NCP SNETMASK definition.

address_mask is a 4-byte address that can be specified in either dotted decimal or hexadecimal notation. All bits in the network portion of the SNETMASK must be on. For class A, B, and C networks, note the following conventions:

Class A   The first 8 bits must all be on (255.n1.n2.n3).
Class B   The first 16 bits must all be on (255.255.n2.n3).
Class C   The first 24 bits must all be on (255.255.255.n3).

The remainder of the SNETMASK value identifies the host and subnetwork addresses where you set bits on to identify the subnetwork address portion and set bits off to identify the host address portion.

To code the address as a dotted decimal number, use the format SNETMASK=*n0.n1.n2.n3* where *n0–n3* are numbers from 0 to 255. For example, code SNETMASK=255.255.240.0.

To code the address as a hexadecimal number, use the format SNETMASK=*hhhhhhhh*. For example, code the address 255.255.240.0 as SNETMASK=FFFFF000.

Note: Avoid defining a subnetwork mask that causes the host address to contain all zeros or all ones. NCP treats datagrams with a resulting destination host address of all zeros as a network address, and all ones as a broadcast address, and discards them.

IPOWNER Definition Statement

Usage: Required for IP dynamic routing. Identifies the IBM TCP/IP MVS/VM host [4] that will manage the NCP internet route table. Also used for dynamic reconfiguration of frame-relay internet hosts.

Position: Definition statements used to define internet routing information to NCP must appear after the last line group and precede the GENEND definition statement in the following order.

1. One IPOWNER definition statement
2. All IPLOCAL definition statements
3. All IPROUTE definition statements

Keywords: Table 2-46 shows the keywords described under this definition statement.

*Table 2-46. Keywords on the IPOWNER Definition Statement*

Keyword	Page	Required	Sift	Default	Tuning*
HOSTADDR	2-161	Yes			
INTFACE	2-161	Yes			
MAXHELLO	2-162			6	
NUMDRIF	2-162			10	
NUMROUTE	2-163			(25,25,25)	
UDPPORT	2-164			580	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature, NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►──┬────────┬──►◄
    └─symbol─┘
```
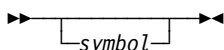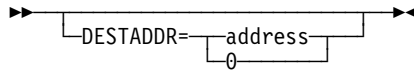
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

---

[4] For information on which product releases and operating systems are supported by NCPROUTE and IBM TCP/IP, see the TCP/IP documentation for your host operating system.

HOSTADDR

►►——HOSTADDR=*host_address*——►◄

Use: Required when you code the IPOWNER definition statement
Default: None

Specifies the internet host address of the owning IBM TCP/IP host.[5] The NCPROUTE program, provided by IBM TCP/IP, manages the NCP internet route table.

The INTFACE name on the IPOWNER definition statement matches the INTFACE name on an IPLOCAL definition statement for the NCST LU or an IP channel connection. The HOSTADDR address should match the P2PDEST address on the same IPLOCAL definition statement.[67]

host_address is a 4-byte internet address that can be specified in either dotted decimal or hexadecimal format.

To code the address as a dotted decimal number, use the format HOSTADDR=*n0.n1.n2.n3* where *n0–n3* are numbers between 0 and 255. For example, code HOSTADDR=128.81.20.1.

To code the address as a hexadecimal number, use the format HOSTADDR=*hhhhhhhh*. For example, code the address 128.81.20.1 as HOSTADDR=80511401. See the *NCP, SSP, and EP Resource Definition Guide* for more information about internet addresses.

INTFACE

►►——INTFACE=(——*name*——)——►◄

Use: Required when you code the IPOWNER definition statement
Default: None

Specifies the name of the internet route interface used to route IP datagrams to the owning IBM TCP/IP host.[8] The name must be from 1 to 8 characters long and must also be the same as the name specified by INTFACE on an NCST LU definition statement or on an IP channel interface definition.

Multiple interfaces can be specified if they are all IP channel interfaces. The first name is used for the primary interface; any following names are for backup interfaces.

---

[5] For information on which product releases and operating systems are supported by NCPROUTE and IBM TCP/IP, see the TCP/IP documentation for your host operating system.

[6] If multiple IP channel interfaces are specified, this applies only to the primary interface.

[7] This restriction does not apply in the absence of SNALINK, where the remote NCP definition must use an indirect path to NCPROUTE via an NCST connection to a local NCP that has an IP channel connection to the NCPROUTE host.

[8] For information on which product releases and operating systems are supported by NCPROUTE and IBM TCP/IP, see the TCP/IP documentation for your host operating system.

MAXHELLO

```
              ┌─MAXHELLO=6──────────┐
►►────────────┼─────────────────────┼──────►◄
              └─MAXHELLO=─┬─count─┬──┘
                          └─NONE──┘
```
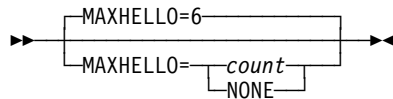
Default: 6
Minimum: 1
Maximum: 2880

Specifies how often NCP attempts to contact NCPROUTE.[9] NCP sends one *hello* datagram every 30 seconds until it receives an acknowledgment. If the maximum number of hellos is sent without an acknowledgment, NCP sends a generic alert and pauses for a 9-minute recovery interval before it begins sending again.

Notes:

1. If you code MAXHELLO=NONE, NCP sends hello datagrams indefinitely and no generic alert is issued.

2. If you code multiple INTFACE names, MAXHELLO=NONE is not valid.

NUMDRIF

```
              ┌─NUMDRIF=10───┐
►►────────────┼──────────────┼──────►◄
              └─NUMDRIF=nnn──┘
```
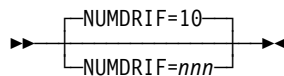
Use: Frame-relay internet devices
Default: 10
Minimum: 0
Maximum: 100

Specifies the number of frame-relay internet interfaces that can be dynamically added to the network.

Note: You can define part of the NCP buffer pool to be used for dynamically allocated internet interfaces. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. Because overuse of dynamic resources can degrade NCP performance, define all the resources required for normal processing in your generation definition and use this function only to provide control block resources for unexpected demand. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

Note: When coding NUMDRIF, remember that excessive values can waste NCP storage.

---

[9] NCPROUTE is a function of IBM TCP/IP that manages the NCP route table.

NUMROUTE
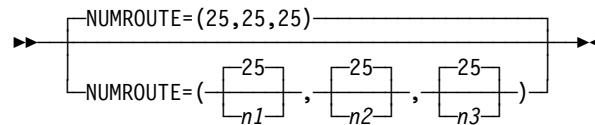
```
►►─┬─NUMROUTE=(25,25,25)──────────────────────┬─►◄
   │              ┌─25─┐     ┌─25─┐     ┌─25─┐  │
   └─NUMROUTE=(───┴─n1─┴──,──┴─n2─┴──,──┴─n3─┴─)┘
```

Specifies how many additional internet route table entries are to be made available for management by the NCPROUTE program.

Note: When coding NUMROUTE, remember that excessive values can waste NCP storage.

n1

Default:	25
Minimum:	0
Maximum:	1000 (for SSP V4R2 and earlier)
	2500 (for SSP V4R3 and later)

Specifies the number of additional resources for dynamically adding routes to an internet host.

n2

Default:	25
Minimum:	0
Maximum:	1000 (for SSP V4R2 and earlier)
	2500 (for SSP V4R3 and later)

Specifies the number of additional resources for dynamically adding routes to an internet subnetwork.

n3

Default:	25
Minimum:	0
Maximum:	1000 (for SSP V4R2 and earlier)
	2500 (for SSP V4R3 and later)

Specifies the number of additional resources for dynamically adding routes to an internet network.

Note: You can define part of the NCP buffer pool to be used for dynamically allocated internet route table entries when explicitly defined internet route table entries have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. Since overuse of dynamic resources can degrade NCP performance, define all the resources required for normal processing in your generation definition and use this function only to provide control block resources for unexpected demand. See the *NCP, SSP, and EP Resource Definition Guide* for more information about dynamically created control blocks.

UDPPORT

```
                       ┌─UDPPORT=580─┐
  ►►──┬─────────────────┬──►◄
      └─UDPPORT=port─┘
```
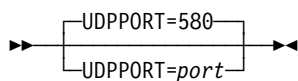
Default: 580
Minimum: 1
Maximum: 65535

Specifies the User Datagram Protocol (UDP) port number used for communication between the NCP internet router and the NCPROUTE server of the owning IBM TCP/IP host.[10] The port number specified must match the port number used by the NCPROUTE program.

---

[10] For information on which product releases and operating systems are supported by NCPROUTE and IBM TCP/IP, see the TCP/IP documentation for your host operating system.

IPROUTE Definition Statement

Usage: For IP. Defines an entry in the internet route table.

Do not code an IPROUTE definition statement for a route that is implicitly defined by an IPLOCAL definition statement.

Position: Definition statements used to define internet routing information to NCP must appear after the last GROUP definition statement and precede the GENEND definition statement in the following order:

1. One IPOWNER definition statement
2. All IPLOCAL definition statements
3. All IPROUTE definition statements

Keywords: Table 2-47 shows the keywords described under this definition statement.

Table 2-47. Keywords on the IPROUTE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
DESTADDR	2-166	Yes			
DISP	2-167				
HOSTRT	2-167				
INTFACE	2-167	Yes			
METRIC	2-168	Yes			
NEXTADDR	2-168			0	
SNETMASK	2-169				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►─────────────────►◄
    └─symbol─┘
```
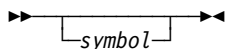
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

DESTADDR

```
►►──────────────────────────────────►◄
      └─DESTADDR=─┬─address─┬─
                  └─0───────┘
```

Use: Required when you code the IPROUTE definition statement
Default: None

Specifies the internet address of the network, subnetwork, or host destination for this internet route.

address is a 4-byte internet address that can be specified in either dotted decimal or hexadecimal format.

To code the address as a dotted decimal number, use the format DESTADDR=*n0.n1.n2.n3* where *n0–n3* are numbers between 0 and 255. For example, code DESTADDR=128.255.0.0.

To code the address as a hexadecimal number, use the format DESTADDR=*hhhhhhhh*. For example, code the address 128.255.0.0 as DESTADDR=80FF0000.

If you specify DESTADDR=0, this statement defines the default route entry. IP datagrams with destination addresses that do not match any of the network or subnetwork destinations defined in the IP route table are routed to this default destination.

Notes:

1. You can define only one default route entry per interface.

2. If HOSTRT=YES, the DESTADDR value must be a host address; if HOSTRT=NO, the DESTADDR value must be a network or subnetwork address.

3. If the DESTADDR value is a subnetwork destination address, a valid subnetwork mask must be defined for this internet address. (See SNETMASK on the IPLOCAL definition statement on page 2-158 or SNETMASK on the IPROUTE definition statement on page 2-169.) A subnetwork of 0 is not valid. Coding zeroes in the subnetwork address portion and the subnetwork host portion causes the address to be interpreted as a network address. The subnetwork specified must be a subnet of a network known to the NCP when VSUBNETS=NO on the BUILD statement. See the *NCP, SSP, and EP Resource Definition Guide* for more information about internet addresses.

4. Do not code the same DESTADDR value on more than one IPROUTE definition statement unless you specify DISP=PERM on each statement.

5. The DESTADDR value cannot be the same as the destination of an implicitly defined route unless DISP=PERM is specified on the IPROUTE definition statement.

6. If the NEXTADDR keyword is omitted or has a value of zero, and the INTFACE value on the IPROUTE definition statement matches the INTFACE value on an IPLOCAL definition statement, the values of LADDR on the same IPLOCAL definition statement and DESTADDR should be in the same network or subnetwork. Otherwise, NDF issues an informational message.

DISP
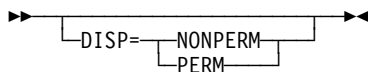
```
►►──┬────────────────────┬──►◄
    └─DISP=─┬─NONPERM─┬───┘
            └─PERM────┘
```

Default: See text

Specifies whether the route being defined by this IPROUTE definition statement is dynamic (NONPERM) or permanent (PERM).

If the values of INTFACE and DESTADDR on the IPROUTE definition statement match the values of INTFACE and HOSTADDR on the IPOWNER definition statement, the default is DISP=PERM and DISP=NONPERM is not valid. Otherwise, the default is DISP=NONPERM.

When you code DISP=PERM, the NCPROUTE program discards all information received about this route; when you code DISP=NONPERM, the NCPROUTE program uses RIP to manage the route.

HOSTRT
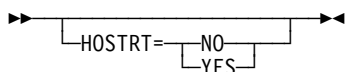
```
►►──┬────────────────────┬──►◄
    └─HOSTRT=─┬─NO──┬─────┘
              └─YES─┘
```

Default: See text

Specifies whether this IPROUTE definition statement defines a route to a specific internet host (HOSTRT=YES), or to an internet network or subnetwork (HOSTRT=NO).

If the values of INTFACE and DESTADDR on this IPROUTE definition statement match the values of INTFACE and HOSTADDR on the IPOWNER definition statement, the default is HOSTRT=YES, and HOSTRT=NO is not valid. Otherwise, the default is HOSTRT=NO.

If DESTADDR=0 is coded, only HOSTRT=NO is valid on this IPROUTE definition statement.

INTFACE

```
►►──INTFACE=name──►◄
```

Use: Required when you code the IPROUTE definition statement
Default: None

Specifies the name of the internet interface used to route IP datagrams to the network, subnetwork, or host destination defined by DESTADDR on this IPROUTE definition statement. The name must be 1 to 8 characters long and must be the same as the value specified by one of the following:

• The INTFACE keyword on the LINE definition statement that defines a NTRI or Ethernet-type LAN attachment

• The INTFACE keyword on the LU definition statement that defines an NCST session interface

- The INTFACE keyword on the PU definition statement that defines a frame-relay interface, channel interface, or internal NCP to Model 900 point-to-point IP connection

METRIC

```
►►──METRIC=value──►◄
```

Use: Required when you code the IPROUTE definition statement
Default: None
Minimum: 1
Maximum: 15

For non-permanent routes, specifies a metric value used by NCPROUTE to evaluate new route information received in RIP packets. For permanent routes, specifies a metric value used by NCPROUTE to determine route selection order.

If the values coded for INTFACE and P2PDEST on the IPLOCAL definition statement match the values coded for INTFACE and HOSTADDR on the IPOWNER definition statement, and an IPROUTE definition statement is coded with the same values for INTFACE and DESTADDR, respectively, NCP will use the METRIC value coded on the IPROUTE definition statement when the route is built.

NEXTADDR

```
        ┌─NEXTADDR=0───────┐
►►──────┤                  ├──►◄
        └─NEXTADDR=address─┘
```

Default: 0

Specifies the address of the next internet gateway to which an IP datagram is to be routed on its way to the destination network or subnetwork, or the internet address of the NCST logical unit session partner in a point-to-point connection. For NCST interfaces, the address specified must be in the same network or subnetwork as the local address specified by the INTFACE keyword on this IPROUTE definition statement.

address is a 4-byte internet address that can be specified in either dotted decimal or hexadecimal format.

To code the address as a dotted decimal number, use the format NEXTADDR=*n0.n1.n2.n3* where *n0–n3* are numbers between 0 and 255. For example, code NEXTADDR=128.255.16.33.

To code the address as a hexadecimal number, use the format NEXTADDR=*hhhhhhhh*. For example, code the address 128.255.16.33 as NEXTADDR=80FF1021. See the *NCP, SSP, and EP Resource Definition Guide* for more information on internet addresses.

Notes:

1. For IP over frame relay:

   - Keywords on the IPLOCAL definition statement can be specified on the PU definition statement.

2. For each IPROUTE definition statement:

   - If the NEXTADDR keyword is omitted or has a value of zero, and the INTFACE value on the IPROUTE definition statement matches the INTFACE value on an IPLOCAL definition statement, the values of LADDR on the same IPLOCAL definition statement and DESTADDR should be in the same network or subnetwork. Otherwise, NDF issues an informational message.

   - If the INTFACE value on the IPROUTE definition statement matches the INTFACE value on an IPLOCAL definition statement, the LADDR value on the same IPLOCAL definition statement and the nonzero NEXTADDR value should be in the same network or subnetwork. Otherwise, NDF issues an informational message.

   - If the INTFACE value on the IPROUTE definition statement matches the INTFACE value on an IPLOCAL definition statement and this is an NCST interface, the P2PDEST value on the same IPLOCAL definition statement and the nonzero NEXTADDR value should be the same. Otherwise, NDF issues an informational message.

3. If several IPROUTE definition statements are coded with the same INTFACE value and this is an NCST interface, all nonzero NEXTADDR values on these definition statements must be the same.

4. Although the NEXTADDR keyword is optional for NCST interfaces, if you define multiple routes that designate the same NCST interface as the next hop, code the same NCST interface value on the INTFACE keyword and code the same nonzero address on the NEXTADDR keyword on each IPROUTE definition statement.

SNETMASK

```
►►──┬──────────────────────────┬──►◄
    └─SNETMASK=address_mask─┘
```
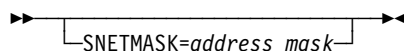
Use: Internet Protocol subnetworks
Default: See text

Specifies the subnetwork mask to associate with the destination address (DESTADDR) when defining a subnetwork route. The subnetwork mask determines which portion of an internet address, defined by the DESTADDR keyword, identifies the subnetwork. The SNETMASK value must contain the network portion of the internet address defined by the DESTADDR keyword.

When VSUBNETS=NO on the BUILD statement:
        The SNETMASK keyword is not allowed on the IPROUTE statement.

When VSUBNETS=YES on the BUILD statement:
        The SNETMASK keyword specifies the subnet mask to be used for this route. If you omit SNETMASK from the IPROUTE statement, then the IPLOCAL subnet mask, if coded, is used when the destination address

(DESTADDR) is in the same network as the associated interface (INTFACE).

address_mask is a 4-byte address that can be specified in either dotted decimal or hexadecimal notation. All bits in the network portion of the SNETMASK must be on. For class A, B, and C networks, note the following conventions:

Class A   The first 8 bits must all be on (255.n1.n2.n3).
Class B   The first 16 bits must all be on (255.255.n2.n3).
Class C   The first 24 bits must all be on (255.255.255.n3).

The remainder of the SNETMASK value identifies the host and subnetwork addresses where you set bits on to identify the subnetwork address portion and set bits off to identify the host address portion.

To code the address as a dotted decimal number, use the format SNETMASK=*n0.n1.n2.n3* where *n0–n3* are numbers between 0 and 255. For example, code SNETMASK=255.255.240.0.

To code the address as a hexadecimal number, use the format SNETMASK=*hhhhhhhh*. For example, code the address 255.255.240.0 as SNETMASK=FFFFF000.

Note: Avoid defining a subnetwork mask that causes the host address to contain all zeros or all ones. NCP treats datagrams with a resulting destination host address of all zeros as a network address, and all ones as a broadcast address, and discards them.

LINE Definition Statement

Usage: Required. One for each telecommunication link in the network.

Position: Can follow COMP, GROUP, LU, PU, TERMINAL, or other LINE definition statement.

Keywords: Table 2-48 shows the keywords described under this definition statement.

Table 2-48 (Page 1 of 5). Keywords on the LINE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ADAPTER	2-179		G	TIC1	
ADDIFG	2-180		G	NO	
ADDRESS	2-180				
ANS	2-193		G		
ANSTONE	2-193		G	NO	Yes*
ARPTAB	2-194				
AUTO	2-194				
AUTOCOPY	2-196				
AUTODL	2-198			YES	
AUTOLINE	2-199				
AUTUACB	2-200		G		
AVGPB	2-200		G		
BALANCE	2-200		G	0	Yes*
BEACTO	2-201		G	52	
BUFETTE	2-201			2	
BUFSIZE	2-201			64	
CA	2-202		G		
CALINE	2-202		G		
CALL	2-203		G	IN	
CASDL	2-203		G		
CCLID	2-204		G	NO	Yes*
CHANLA	2-204				
CHECK	2-205			NODCD	
CHNLZ	2-205		G	NO	
CHNPRI	2-206			NORMAL	
CIR	2-206		G	NO	Yes*

Sift:

B   BUILD definition statement
G   GROUP definition statement

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-48 (Page 2 of 5). Keywords on the LINE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
CLINES	2-207				
CLLM	2-207		G	NO	Yes*
CLOCKNG	2-208		G		Yes*
CODE	2-210		G		
COMRATE	2-211		G		Yes*
CONFIG	2-212		G	NONSW	
CORNUM	2-213				
CRATEDLY	2-214		G		Yes*
CRRATE	2-214				
CSPMODE	2-215				
CU	2-215			2703	
CUTOFF	2-215		G	NO	
CUTYPE	2-216		G	3271	
DATABLK	2-217		G	2048	
DATRATE	2-217		G	LOW	Yes*
DELAY	2-217		G	0.1	
DIALALT	2-218		G	NONE	
DIALSET	2-218		G	NONE	
DISABLE	2-218			NO	
DSCRDDLY	2-219		G		Yes*
DUALCOM	2-219			NONE	
DUPLEX	2-220		G	HALF	Yes*
DYNADMP	2-221			NONE	
DYNWIND	2-222			(4,1,1,1,Yes)	Yes*
ETRATIO	2-224		G	30	Yes*
FEATURE	2-224		G		
FRAMECNT	2-226			(100000,5000)	
FRSPRI	2-227		G	0	Yes*
FRRATES	2-227		G	(1,1,1)	Yes*
HDXSP	2-228		G	NO	Yes*
HICHAN	2-228			NONE	
HISPEED	2-229		G		Yes*
HLOQSTOP	2-229		G		Yes*

Sift:

B   BUILD definition statement
G   GROUP definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-48 (Page 3 of 5). Keywords on the LINE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
HOSTLINK	2-229				
ID	2-230		G		
INBFRS	2-230		G	3	
INTFACE	2-231				
IPL	2-232		G	NO	
ISDNCALL	2-232			OUT	
ISDNLPN	2-233				
ISDNLSA	2-234				
ISDNRPN	2-235				
ISDNRSA	2-235				
LIMRES	2-236		G	NO	
LINECB	2-236		G		
LINEFVT	2-236		G		
LINESIZ	2-237		G		
LNQTCNT	2-237		G	1	
LOCADD	2-238				Yes*
LOCALTO	2-239		B,G		Yes*
LOCHAN	2-240			NONE	
LPDATS	2-240		G	NO	
LPDA2DL	2-243		G	NO	
LSPRI	2-244			PU	Yes*
MAXDLCI	2-245				
MAXFRAME	2-246		G	2106	
MAXPU	2-248		G		
MAXTSL	2-249		G		
MLTGPRI	2-250		G	0	Yes*
MODEM	2-250			OPTION2	
MONITOR	2-251		G	NO	
MONLINK	2-252		G		
MPTALT	2-253		G	NO	
MTALIST	2-254		G		
MULTI	2-254			NO	
NCPCA	2-254		G	ACTIVE	

Sift:

B   BUILD definition statement
G   GROUP definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-48 (Page 4 of 5). Keywords on the LINE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
NEGPOLP	2-255		G	NONE	
NEWSYNC	2-255		G	NO	Yes*
NPACOLL	2-256		G		
NPATP	2-257		G	NO	
NRZI	2-258		G		Yes*
PAD	2-259			YES	
PARCHK	2-259		G	NOCHECK	
PARGEN	2-259		G	EVEN	
PAUSE	2-260		G	(0.2,2.8)	Yes*
POLIMIT	2-261		G	(1,NOWAIT)	
POLLED	2-262		G	NO	
POLLTO	2-263		G	ERROR	
PORT	2-263				
PORTADD	2-263				
PROMPT	2-265		G	YES	
QUIET	2-265			NO	
QSIZE	2-265		G		Yes*
QSIZEDE	2-266		G		Yes*
RCVBUFC	2-266		G		
RCVCHAR	2-267			4	
REDIAL	2-267		G		Yes*
REMOTTO	2-268		B,G		Yes*
RETRIES	2-269		G		Yes*
RING	2-272		G	NO	Yes*
SDLCST	2-273		G		
SECURE	2-273		G	NO	
SERVLIM	2-274		G		Yes*
SERVPRI	2-275		G	OLD	
SESSION	2-275		G	1	
SPDSEL	2-276		G	NO	
SPEED	2-276		G		Yes*
SPSHIFT	2-279		G	NO	
SRETRY	2-280		G	15	Yes*

Sift:

B   BUILD definition statement

G   GROUP definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
       NTuneMON V2R5 User's Guide for more information.

Table 2-48 (Page 5 of 5). Keywords on the LINE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
SWMMTID	2-280		G		
TADDR	2-280		G		
TAILING	2-282		G	NO	
TERM	2-282		G		
TIMEOUT	2-284		G		
TITIMER	2-285			60	Yes*
TRANSBUF	2-285				Yes*
TRANSFR	2-286		G		Yes*
TRSPEED	2-288		G		
TYPE	2-288				
T1TIMER	2-289		B,G		Yes*
T2TIMER	2-291		G		Yes*
UACB	2-292				
UNITXC	2-293			YES	
USE	2-293		G	NCP	Yes*
V25BIS	2-294		G	NO	
YIELD	2-295		G	YES	

Sift:

B   BUILD definition statement
G   GROUP definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements:
You can define most keywords of the CLUSTER, COMP, LU, PU, and TERMINAL definition statements on the LINE definition statement (or the GROUP definition statement). Additionally, you can code most LINE keywords on the GROUP definition statement. Table 1-3 on page 1-12 shows which lower-level keywords you can code at a higher level.

VTAM Keywords: The following diagrams show the VTAM keywords that can be coded on this definition statement. These keywords provide information to the VTAM initialization process. They are not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for descriptions of these keywords and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

VTAM keywords not listed below can be specified by coding VTM. as a prefix in front of the VTAM keyword.

```
      ┌─ANSWER=ON─┐
►►─────┴─ANSWER=─┬─ON──┬─┴──────►◄
                 └─OFF─┘
```

```
►►─┬──────────────────────────────────────────────┬──►◄
   └─ASLENT=associated_LU_table_entry_name─────────┘


►►─┬──────────────────────────────────────┬──►◄
   └─ASLTAB=associated_LU_table_name───────┘


►►─┬────────────────────┬──►◄
   └─CAPACITY=─┬─nnnK─┬──┘
               └─nnnnM─┘


    ┌─CONNTYPE=APPN──────────┐
►►──┼────────────────────────┼──►◄
    └─CONNTYPE=─┬─APPN─┬──────┘
               └─LEN──┘


►►─┬───────────────┬──►◄
   └─COSTBYTE=n─────┘


►►─┬───────────────┬──►◄
   └─COSTTIME=n─────┘


►►─┬─────────────┬──►◄
   └─CPCP=─┬─YES─┬─┘
          └─NO──┘


    ┌─DISCNT=NO────────────────────────────────┐
►►──┼──────────────────────────────────────────┼──►◄
    └─DISCNT=─┬─(─YES─┬─────────┬──┬──────────┬─)─┘
             │       └─,─F─────┘  │          │
             │       ┌─,─┬─F──┬───┘          │
             │       └───┴─NF─┘              │
             └─NO──────────────────────────────┘


►►─┬───────────────────────────────────────────┬──►◄
   └─DLOGMOD=default_logon_mode_entry_name──────┘


►►─┬──────────────────┬──►◄
   └─DYNADJCP=─┬─YES─┬──┘
             └─NO──┘


►►─┬───────────────┬──►◄
   └─DYNLU=─┬─YES─┬──┘
          └─NO──┘


    ┌─DYNPU=NO──────────┐
►►──┼───────────────────┼──►◄
    └─DYNPU=─┬─NO──┬─────┘
           └─YES─┘


    ┌─DYNPUPFX=CN──────────┐
►►──┼──────────────────────┼──►◄
    └─DYNPUPFX=string───────┘


    ┌─ENCR=NONE────────────┐
►►──┼──────────────────────┼──►◄
    └─ENCR⁽¹⁾=─┬─NONE─┬──────┘
             ├─REQD─┤
             ├─SEL──┤
             └─OPT──┘
```

[1] Valid for MVS only

```
►►──┬─────────────────────────────┬──►◄
    └─FEATUR2=(──▼─keyword──┬─)──┘
                  └───,─────┘
```

```
►►──┬─────────────┬──►◄
    └─HPR=──┬─YES─┬─┘
           └─NO──┘
```

```
►►──┬───────────────────┬──►◄
    └─INCLUD0E=──┬─NO──┬─┘
                └─YES─┘
```

```
         ┌─ISTATUS=ACTIVE────────────┐
►►──┬────┴──────────────────────────┴──►◄
    └─ISTATUS=──┬─ACTIVE───┬──┘
               └─INACTIVE─┘
```

```
►►──┬───────────────────────────────────┬──►◄
    └─LOGAPPL=application_program_name──┘
```

```
►►──┬─────────────────────────────┬──►◄
    └─LOGTAB=interpret_table_name─┘
```

```
►►──┬──────────────────────────────────┬──►◄
    └─MDLENT=model_name_table_entry_name─┘
```

```
►►──┬──────────────────────────────┬──►◄
    └─MDLTAB=model_name_table_name─┘
```

```
►►──┬──────────────────────────────┬──►◄
    └─MODETAB=logon_mode_table_name─┘
```

```
►►──┬────────────────┬──►◄
    └─NATIVE=──┬─NO──┬─┘
             └─YES─┘
```

```
►►──┬────────────┬──►◄
    └─NN=──┬─YES─┬─┘
         └─NO──┘
```

```
►►──┬─────────────────────────────┬──►◄
    └─OWNER=(──▼─owner_name──┬─)──┘
                 └───,───────┘
```

```
    ┌─PDELAY=1─┐
►►──┼──────────┼──►◄
    └─PDELAY=n─┘
```

```
    ┌─PU=YES────┐
►►──┼───────────┼──►◄
    └─PU=──┬─YES─┬─┘
         └─NO──┘
```

```
►►──┬──────────────────────┬──►◄
    └─REGISTER=──┬─NO──────┬─┘
               ├─NETSRVR─┤
               └─DIRSRVR─┘
```

LINE

```
           ┌─SECNET=NO────────┐
   ▶▶──────┴──────────────────┴──▶◀
            └─SECNET=─┬─NO──┬─┘
                      └─YES─┘


   ▶▶───────┬──────────────────────┬──▶◀
            └─SECURITY=─┬─UNSECURE─┬─┘
                        ├─PUBLIC───┤
                        ├─UNDERGRO─┤
                        └─SECURE───┘


   ▶▶───────┬───────────────────┬──▶◀
            └─SPAN=span_name────┘


           ┌─SSCPFM=FSS──────────┐
   ▶▶──────┴─────────────────────┴──▶◀
            └─SSCPFM=─┬─FSS─────┬─┘
                      ├─USSSCS──┤
                      ├─USS3270─┤
                      ├─USS3275─┤
                      ├─USS3780─┤
                      └─USSNTO──┘


   ▶▶───────┬─────────────┬──▶◀
            └─TGP=tgp_name─┘


           ┌─TOPO=CONNECT──────┐
   ▶▶──────┴───────────────────┴──▶◀
            └─TOPO=─┬─CONNECT─┬─┘
                    └─NONE────┘


   ▶▶───────┬─────────┬──▶◀
            └─UPARM1=n─┘


   ▶▶───────┬─────────┬──▶◀
            └─UPARM2=n─┘


   ▶▶───────┬─────────┬──▶◀
            └─UPARM3=n─┘


   ▶▶───────┬───────────────────────────────┬──▶◀
            └─USSTAB=USS_definition_table_name─┘


           ┌─VERSALID=NO───────┐
   ▶▶──────┴───────────────────┴──▶◀
            └─VERSALID=─┬─NO──┬─┘
                        └─YES─┘


   ▶▶───────┬───────────────────────┬──▶◀
            └─VNGROUP=group_name─────┘


   ▶▶───────┬──────────────────────────────────┬──▶◀
            └─VNNAME=connection_network_cpname──┘


   ▶▶──VTAMTOPO=─┬─NOREPORT─┬──▶◀
                 └─REPORT───┘
```

Keyword Descriptions

symbol

►►──*symbol*──►◄

| Use: Required for NCP and EP lines
   Default: None

Provides a name for the communication line. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The *symbol* appears in the label cross-reference listing that the generation process creates. If AUTOLINE is specified, you must embed the first suboperand of the ADDRESS keyword on the LINE definition statement in the symbol.

ADAPTER

```
            ┌─ADAPTER=TIC1─────────┐
►►─────────┤                      ├──►◄
            └─ADAPTER=──┬─TIC1─┬──┘
                        ├─TIC2─┤
                        └─TIC3─┘
```
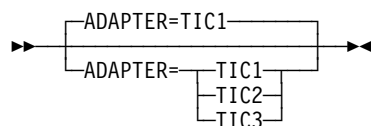
Use: Token-Ring physical resources
Default: TIC1

Specifies which type of TIC is attached to this physical line.

TIC1│**TIC2**

Use: NTRI

Specifies whether this line is attached to a type 1 TIC or a type 2 TIC. Do not code ADAPTER=TIC1 if you code ECLTYPE=(PHYSICAL,ANY) on the GROUP definition statement because the type 1 TIC does not support a mixture of subarea and peripheral connections. Do not code ADAPTER=TIC1 or ADAPTER=TIC2 if you code ADAPTER=TIC3 on the GROUP definition statement.

TIC3

Use: 3746 Model 900 Token Ring

Specifies that this line is attached to a type 3 TIC. Code ADAPTER=TIC3 on the GROUP definition statement to define 3746 Model 900 Token-Ring physical lines.

The value of ADAPTER affects the default for ECLTYPE on the GROUP definition statement. See page 2-115 for more information.

ADDIFG

```
                   ┌─ADDIFG=NO─────┐
►►─────────────────┴───────────────┴────►◄
         └─ADDIFG=─┬─YES─┬─
                   ├─NO──┤
                   └─nn──┘
```
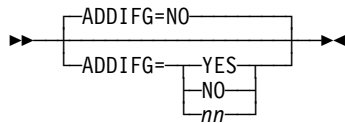
Use: 3746 Model 900 frame-relay and 3746 Model 900 SDLC physical lines
Default: NO

Specifies whether the gap between frames should be made larger than a minimum
of 1 flag between frames, and if so, how much larger. Larger interframe gaps
cause the frame rate to decrease, which helps to keep the adjacent adapter from
getting overrun errors. Specifying a larger gap is recommended when the adjacent
adapter to which this adapter is transmitting is experiencing overrun errors.

Note: To use the ADDIFG keyword when generating NCP V7R1, you must install
NCP APAR IR27090. For V6R3 you must install NCP APAR IR27091. *nn*
is valid starting with NCP V7R3.

NO
Specifies that the 3746 Model 900 transmit a minimum of 1 flag between
frames.

YES
Specifies a line-speed-dependent value for the minimum number of flags that
the 3746 Model 900 transmits between frames. The line speed determines the
minimum number of flags as follows:

256000 bps or greater: `minimum flags = line speed / 8500`
255999—25600 bps: `minimum flags = line speed / 2560`
25600 bps or less: `minimum flags = 10`

nn

Minimum: 1
Maximum: 255

Specifies a value (*nn*) for the minimum number of flags that the 3746 Model
900 transmits between frames.

ADDRESS
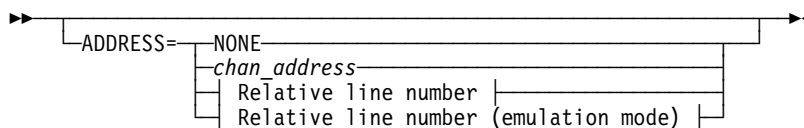
```
►►──┬─────────────────────────────────────────────────┬────►◄
    └─ADDRESS=─┬─NONE──────────────────────────────┬─
               ├─chan_address──────────────────────┤
               ├─┤ Relative line number ├──────────┤
               └─┤ Relative line number (emulation mode) ├─┘
```

Relative line number:
```
            ┌─,HALF────┐
├──(─lnbr─┬─┴─────────┴─┬──)──┤
          └─,─┬─HALF─┬──┘
              └─FULL─┘
```

Relative line number (emulation mode):
```
              ┌─;──────────────┐
├──(lnbr,─┴─▼─subch_address─────┴───────┬──)──┤
                              └─-ca_address─┘
```
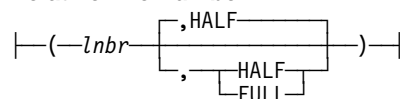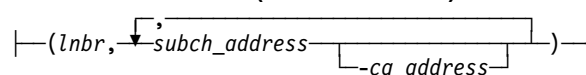
Use: Physical links and ISDN B-channel links

Examples: ADDRESS=7
                 ADDRESS=(21,4F-P5,E8-P1) (emulation subchannel)

Specifies the relative line number and data transfer mode, or emulation subchannel addresses, for this line. ADDRESS also defines the channel adapter logical address for user-written channel code, NCP subarea channel links, and NCP peripheral channel links.

ADDRESS is optional if you specify LINEADD=NONE or VIRTUAL=YES on the GROUP definition statement.

Note: There are special channel adapter assignment requirements for IBM 3745-410, 3745-41A, 3745-610, and 3745-61A Communication Controllers operating in twin-in-dual mode. See Table 2-51 on page 2-185 for more information.

Spare or Redefinable Lines: Spare or redefinable lines are supported only on TSS and HPTSS adapters. Therefore, the ADDRESS value specified must be valid for TSS-attached or HPTSS-attached lines.

NONE

Specifies no addresses for the line represented by this LINE definition statement. Code ADDRESS=NONE if you code LEVEL2 and VIRTUAL=NO on the GROUP definition statement.

3746 Model 900 Token Ring: Code ADDRESS=NONE for logical lines.

3746 Model 900 Frame Relay: ADDRESS=NONE is the only valid value for logical lines.

ESCON Links: Omit ADDRESS or code ADDRESS=NONE for logical lines.

ISDN: ADDRESS=NONE is the only valid value when you specify ISDN=(LOGICAL,SUBAREA) or ISDN=(LOGICAL,PERIPHERAL).

chan_address

Default: None

Specifies the channel adapter address for this line. You can specify the channel adapter address by logical address or physical position. For more information, see "Channel Adapter Addresses for the IBM 3745 Communication Controller" on page 2-184.

Note: If your channel adapter has a two-processor switch (CA=TYPE*n*-TPS on this LINE definition statement), specify an even number for the logical address or an odd number following *P* for the physical position.

Subarea and Peripheral Channel Links: Specifies the channel adapter address if this LINE definition statement is in a line group for which LNCTL=CA is defined. For information on specific channel numbers, see Table 2-50 on page 2-184.

ESCON Links: ESCON physical links are identified by relative line number. See "Relative Line Numbers for the 3746 Model 900 Connectivity Subsystem" on page 2-188 for more information.

User Subchannels: Specifies the channel adapter address if this line is in a group for which CHANLNK=YES is specified. If you define more than one channel adapter with user subchannels, they can be in the same or

different line groups, but you must specify different channel adapter addresses on the ADDRESS keyword of each LINE definition statement. See the description of UCHAN on the BUILD definition statement.

lnbr

Default: None

Specifies the relative line number of the line. For more information on determining the values to code, see "Relative Line Numbers for the IBM 3745 Communication Controller" on page 2-185.

<u>HALF</u>│FULL

Default: HALF

Specifies whether NCP is to use this line in half-duplex or duplex mode.

HALF and FULL are valid only for lines operating in network control mode. Do not specify FULL for a BSC line operating in network control mode. You may specify FULL if you code AUTO on the LINE definition statement.

If you code HALF, NCP converts the relative line number specified in the first parameter to an even address over which data is both transmitted and received. If you code FULL, NCP converts the relative line number to a pair of even and odd addresses. Data is then transmitted over the line by using the even address, and received by using the odd address.

3746 Model 900 Token Ring: Code ADDRESS=(*lnbr*,FULL) for physical lines. Table 2-57 on page 2-190 shows the valid line numbers for physical lines.

3746 Model 900 Frame Relay: Code ADDRESS=(*lnbr*,FULL) for physical lines. Table 2-58 on page 2-192 shows the valid line numbers for 3746 Model 900 frame-relay physical lines.

ISDN: Code ADDRESS=(*lnbr*,FULL) for physical lines. Table 2-58 on page 2-192 shows the valid line numbers for ISDN physical lines. Code the ISDN B-channel addresses in the range of reserved addresses following the address of the associated ISDN physical line.

Table 2-49 on page 2-183 shows the effect on the HALF/FULL suboperand when you code certain keywords on the GROUP definition statement.

Table 2-49. NDF Values for HALF/FULL on ADDRESS on the LINE Definition Statement

When You Code This on GROUP	The Effect on ADDRESS Is
X21NTWK=OLD or 1984	The default is FULL.
ECLTYPE=PHYSICAL	NDF adds FULL. The only valid value is FULL.
ETHERNET=PHYSICAL	The default is FULL. The only valid value is FULL.
FRELAY=PHYSICAL	NDF adds FULL. The only valid value is FULL.
ISDN=PHYSICAL ISDN=(LOGICAL,B)	NDF adds FULL. The only valid value is FULL.

subch address [*-ca address*]

Use: EP
Default: None

Specifies an emulation subchannel address and an optional channel adapter logical address or physical position for the line represented by this LINE definition statement. For more information on determining the values to code, see "Relative Line Numbers for the IBM 3745 Communication Controller" on page 2-185.

When specifying a subchannel address, code a hexadecimal value for *sa* to represent the subchannel address and code a decimal value or physical position for *ca* to indicate the corresponding channel adapter address. Include a hyphen between the hexadecimal value and the decimal value.

Channel adapter address 0 is the default if you do not specify an address. For example, ADDRESS=(21,4F,E8-1) indicates subchannel 4F on the first channel adapter at logical address 0, and subchannel E8 on the second channel adapter at logical address 1.

If you are specifying channel adapter addresses by physical position, ADDRESS=(21,4F-P5,E8-P1) indicates subchannel 4F on the channel adapter in physical position 5 and subchannel E8 on the channel adapter in physical position 1.

Subchannel addresses must be within the range specified by the HICHAN and LOCHAN keywords on either the BUILD or LINE definition statement. The subchannel address specified for this line must not be the same as a subchannel address associated with any other line or used by any other program facility (such as the dynamic dump utility). For additional information on specifying logical addresses, see "Channel Adapter Addresses for the IBM 3745 Communication Controller" on page 2-184.

EP properly initializes its resources when a scanner owning a line and a channel adapter owning the subchannel address for that line are on the same CCU. You must specify the channel adapter in the first position on the ADDRESS keyword if the scanner is also specified on the same CCU. If you do not specify the channel adapter on the same CCU, or if the channel adapter is not operational, the line is not initialized.

If you specify an operational channel adapter on the ADDRESS keyword in a position other than the first, EP uses the primary (first) channel adapter to initialize the line. To properly initialize resources, channel adapters must be specified in the proper position on the ADDRESS keyword.

You may specify uninstalled or disconnected channel adapters in any order or position. You may not specify a channel adapter attached to a CCU in a line definition for any other CCU.

Do not split EP resources across CCU boundaries. You must specify only fallback channel adapters for lines on fallback scanners. When you specify fallback resources, the scanner owning the line and the channel adapter owning the ESC should be on buses attached to the other CCU.

Channel Adapter Addresses for the IBM 3745 Communication Controller

To specify a channel adapter address for an IBM 3745 Communication Controller, you can specify either a physical position or a logical address. Table 2-50 shows the valid channel adapter logical addresses and physical positions for controller models and the correspondence between physical positions and logical addresses.

To specify a physical position, code P*n*, where *n* is the position number. For example, code ADDRESS=P2 to specify physical position 2. If you are defining more than one channel link, you can specify some channel adapter addresses by logical address and others by physical position. However, it is recommended that you use only one method for all channel links.

Table 2-50. Correspondence between IBM 3745 Channel Adapter Physical Positions and Logical Addresses

Bus Position	IBM 3745 Logical Address		Physical ADDRESS Value
	Model 3745-x10 or 3745-x1A	Model 130, 170, or 17A[1]	
1	8[2]		P1
2	9[2]		P2
3	10[2]		P3
4	11[2]		P4
5	0	0	P5
6	1	1	P6
7	2	2	P7
8	3	3	P8
9	12[2]		P9
10	13[2]		P10
11	14[2]		P11
12	15[2]		P12
13	4		P13
14	5		P14
15	6		P15
16	7		P16

[1] In a Model 130 or 170 controller, the four physical channels are attached to bus positions 5, 6, 7, and 8.

[2] In a twin-CCU controller, these channel adapters are assigned to CCU A.

Table 2-51 on page 2-185 lists valid physical positions by usage tier. For more information on usage tiers, see *NCP, SSP, and EP Resource Definition Guide*.

Table 2-51. IBM 3745 Channel Adapter Physical Positions for Usage Tiers

Usage Tier	3745-x10 or 3745-x1A		Model 130, 170, or 17A
	Operating Mode[1]	**Valid Physical Positions: 1–16**	**Valid Physical Positions: 5–8**
1		any 2[5]	any 2[5]
2	S, TS	any 2[6]	any 2[5]
	TB, TD[2]	1 on CCU A,[3,6] 1 on CCU B[4,6]	
2.5	S, TS	any 4[5]	all
	TB, TD[2]	any 2 on CCU A,[3,5] any 2 on CCU B[4,5]	
3	S, TS	any 8[5]	all
	TB, TD[2]	any 4 on CCU A,[3,5] any 4 on CCU B[4,5]	
4		all	all
5		all	not applicable

[1] S=single CCU, TS=twin-in-standby, TB=twin-in-backup, TD=twin-in-dual. See USGTIER on the BUILD definition statement on page 2-59 and the *NCP, SSP, and EP Resource Definition Guide* for more information about these suffixes.

[2] Twin-in-dual mode supports only the channel adapters attached to the CCU that is loaded.

[3] Positions 1–4 and 9–12.

[4] Positions 5–8 and 13–16.

[5] If no transmission subsystem (TSS), high-performance transmission subsystem (HPTSS), Token-Ring (TRA), or Ethernet (ETHR) adapter lines are defined in NCP, you can use double the indicated number of channel adapters (for example, 4 instead of 2).

[6] If no transmission subsystem (TSS), high-performance transmission subsystem (HPTSS), Token-Ring (TRA), or Ethernet (ETHR) adapter lines are defined in NCP, you can use triple the indicated number of channel adapters (for example, 6 instead of 2).

Relative Line Numbers for the IBM 3745 Communication Controller

Relative line numbers identify the port and the line adapter address for the LINE statement. The relative line number used to identify the port and the line adapter are determined by the type of line adapter.

Note: Switched interfaces are supported only on TSS adapters.

Table 2-52 on page 2-186 summarizes the relative line numbers used to identify HPTSS, Token-Ring subsystem (TRSS), and ESS line adapters.

Table   2-52. Relative Line Numbers for IBM 3745 HPTSS, TRSS, and ESS Line Adapters

Line Adapter Position	Port	HPTSS	TRSS	ESS
1	0	1024[1]	1088	1056[1]
	1	1025[1]	1089	1057[1]
2	0	1026[1]	1090[2]	1058[1]
	1	1027[1]	1091[2]	1059[1]
3	0	1028		1060
	1	1029		1061
4	0	1030		1062
	1	1031		1063
5	0	1032[1]	1092[1]	1064[1]
	1	1033[1]	1093[1]	1065[1]
6	0	1034[1]	1094[1]	1066[1]
	1	1035[1]	1095[1]	1067[1]
7	0	1036[1]		1068[1]
	1	1037[1]		1069[1]
8	0	1038[1]		1070[1]
	1	1039[1]		1071[1]

[1] This combination of line adapter position and type is not supported by the IBM 3745-1xx.

[2] This combination of line adapter position and type is not supported by the IBM 3745-160.

Frame Relay: FRELAY=(PHYSICAL,ANY) is only valid for HPTSS-attached lines when the HPTSS adapter is loaded with one of the following microcode levels to support both peripheral and subarea connections:

- *C38006* for the IBM 3745 Models 130, 150, 160, 170
- *C37967* for the IBM 3745 Models 210, 310, 410, 610
- *D39888* for the IBM 3745 Models 21A, 31A, 41A, 61A
- *D400002* for the IBM 3745 Model 17A

The following information provides more detail about the line adapters supported by specific IBM 3745 models and corresponding usage tier levels.

IBM 3745-x10 and 3745-x1A Relative Line Numbers: The IBM 3745 has sepa-
rate relative line number ranges for TSS, HPTSS, ESS, and TRSS adapters.
Table 2-53 shows these ranges for the IBM 3745-x10 and 3745-x1A models.

Table   2-53. Relative Line Numbers for the IBM 3745-x10 and 3745-x1A

Adapter Type	Line Numbers
TSS	0–895[1]
HPTSS	1024–1039
ESS	1056–1071[2]
TRSS	1088–1095[2]

[1] Even though the range of valid line numbers extends to 895, you can define no more than 512 TSS lines.

[2] Specify sequential even-odd pairs of addresses (for example, 1056 and 1057 or 1088 and 1089).

IBM 3745-x10 and 3745-x1A Adapter Positions: Table 2-54 shows the LA,
TRSS adapter, and ESS adapter positions for the IBM 3745-x10 and 3745-x1A
models and how many of each type of adapter you can define. (Line adapters
include TSS and HPTSS adapters.) See the chapter on IBM 3745 support in the
NCP and EP Reference for more information on the correlation between adapter
positions and relative line numbers. See *NCP, SSP, and EP Resource Definition
Guide* for more information on usage tiers.

Table   2-54. Adapter Positions for the IBM 3745-x10 and 3745-x1A

		Positions with TRSS or ESS Adapters	
Usage Tier	LAs without TRSS or ESS Adapters (1–32)	LAs (3, 4, 7–32)	TRSS (1, 2, 5, 6) and ESS Adapters (1–8)
1	1	1	1
2	2	2	1
2.5	6	6[1]	2[2]
3[3]	8	8[1]	4[2]
4[3]	24	24[1]	8[2]
5[3]	32	32[1]	8[2]

[1] You may define LAs for positions 3, 4, 7, and 8 only if there are no ESS adapters occu-
pying those positions.

[2] You may define ESS adapters for positions 3, 4, 7, and 8 only if there are no LAs occu-
pying those positions.

[3] If your IBM 3745 will operate in twin-in-dual or twin-in-backup mode, NCP supports half
the number of adapters on the loaded CCU's nonswitched bus. See the *NCP and EP
Reference* and the *NCP, SSP, and EP Resource Definition Guide* for more information on
twin-CCU operation.

IBM 3745-1xx Relative Line Numbers: The IBM 3745 has separate relative line number ranges for TSS, HPTSS, ESS, and TRSS adapters. Table 2-55 shows these ranges for the IBM 3745-1xx.

Table 2-55. Relative Line Numbers for the IBM 3745-1xx

	Line Numbers	
Adapter Type	**3745-130**	**Model 150, 160, 170, or 17A**
TSS	0–159	0–31
HPTSS	1028–1031	1028–1031
ESS	1060–1063[1]	1060–1063
TRSS	1088–1091[1]	1088–1089

[1] Specify sequential even-odd pairs of addresses (for example, 1060 and 1061 or 1088 and 1089).

IBM 3745-1xx Adapter Positions: Table 2-56 shows the line adapter (LA), TRSS adapter, and ESS adapter positions for the IBM 3745-1xx and how many of each type of adapter you can define. (LAs include TSS and HPTSS adapters.) See the chapter on IBM 3745 support in the *NCP and EP Reference* for more information on the correlation between adapter positions and relative line numbers. See *NCP, SSP, and EP Resource Definition Guide* for more information on usage tiers.

Table 2-56. Adapter Positions for the IBM 3745-1xx

		Positions with TRSS or ESS Adapters	
Usage Tier	**LAs without TRSS or ESS Adapters (3, 4, 9–12)[1]**	**LAs (3, 4, 9–12)[1]**	**TRSS (1 and 2)[1] and ESS Adapters (3 and 4)**
1	1	1	1
2	2	2	1
2.5	6	2	2
3	6	6[2]	4[3]
4	6	6[2]	4[3]

[1] For the IBM 3745-150, 3745-160, 3745-170, and 3745-17A, only positions 1, 3, 4, 9, and 10 are valid.

[2] You may define LAs for positions 3 and 4 only if there are no ESS adapters occupying those positions.

[3] You may define ESS adapters for positions 3 and 4 only if there are no LAs occupying those positions.

Relative Line Numbers for the 3746 Model 900 Connectivity Subsystem

The relative line number for a 3746 Model 900 line is determined by the position, type, and port of the communication adapter that supports the line.

The 3746 Model 900 supports the following communication adapters:

- CLP, which supports as many as four line interface couplers (LICs)

- ESCON processor (ESCP)

- Token-Ring processor (TRP), which supports two TICs

- Controller bus and service processor (CBSP), which combines a controller bus processor (CBP) and a telecommunication service console processor (TSCP)

- Control bus and Token-Ring processor (CBTRP), which combines a CBP and a TRP

Table 2-57 on page 2-190 shows the communication adapters supported at each 3746 Model 900 adapter position and the relative line numbers for ESCP and TRP lines at each 3746 Model 900 adapter position. Table 2-58 on page 2-192 shows the relative line numbers for CLP lines at each position.

Notes:

1. Positions 7–12 are not available in some 3746 Model 900s. See the documentation for your 3746 Model 900 to determine whether these positions are available.

2. Do not mix the definition of 3746 Model 900 lines with other types of lines under a single GROUP definition statement.

LINE

Table 2-57. Relative Line Numbers for 3746 Model 900 ESCP and TRP Positions

Position	Adapter Supported	ESCP Relative Line Number	TRP Relative Line Number	
			Port 0	Port 1
1 [1]	None	—	—	—
2 [2]	CBSP	—	—	2080
3 [3]	CBTRP, CLP, ESCP, TRP	2112	2112	2144
4	CLP, ESCP, TRP	2176	2176	2208
5	CLP, ESCP, TRP	2240	2240	2272
6	CLP, ESCP, TRP	2304	2304	2336
7	CLP, ESCP, TRP	2368	2368	2400
8	CLP, ESCP, TRP	2432	2432	2464
9	CLP, ESCP, TRP	2496	2496	2528
10	CLP, ESCP, TRP	2560	2560	2592
11	CLP, ESCP, TRP	2624	2624	2656
12	CLP, ESCP, TRP	2688	2688	2720
13	CLP, ESCP, TRP	2752	2752	2784
14	CLP, ESCP, TRP	2816	2816	2848
15	CLP, ESCP, TRP	2880	2880	2912
16	CLP, ESCP, TRP	2944	2944	2976
17	CLP, ESCP, TRP	3008	3008	3040
18	CLP, ESCP[4], TRP	3072	3072	3104

[1] Position 1 is reserved for the service and power support (SPS) adapter.

[2] Position 2 is reserved for the CBSP adapter. Although the initial adapter line number begins at base line number 2048, the first 32 line numbers of this position are restricted for connecting the communication controller's CCU to the first CBP. You can attach a Token-Ring line only at relative line number 2080 of adapter position 2.

[3] If you are connecting to an IBM 3745-41A or 3745-61A, position 3 is reserved for the CBTRP adapter. In this case, you can attach a Token-Ring line only at relative line number 2144 using adapter position 3. If you are connecting to an IBM 3745-17A, 3745-21A, or 3745-31A, position 3 may be used for an ESCP, a TRP, or a CLP.

[4] Processor slot 18 cannot be used to load NCP via ESCON.

A 3746 Model 900 CLA is made up of a CLP and up to four LICs. Each LIC supports a maximum of 30 lines.

Odd-numbered adapter positions are associated with adjacent even-numbered adapter positions in pairs. Position 3 is associated with position 4, position 5 with position 6, and so on. Each CLP normally controls the two LICs associated with a single CLP adapter position; however, when the adjacent CLP position is vacant, a single CLP can control the two LICs for each position. This is useful for reducing the number of CLPs. Grouping CLPs into logical pairs is also useful to provide backup support. When one of two CLPs in a logical pair fails, the remaining active CLP can assume control of all four LICs.

The 3746 Model 900 CLP supports the following LIC types:

- LIC 11 (line speeds up to 256 Kbps)
- LIC 12 (line speeds up to 2 Mbps)
- LIC 16 (line speeds up to 2 Mbps)

Table 2-58 on page 2-192 shows the relative line numbers for these LIC types at each CLP position.

Table   2-58. Relative Line Numbers for 3746 Model 900 CLP Positions

Position	Port	LIC 11[1]	LIC 12 or LIC 16
3	0	2112–2126, 2128–2142	2112
	1	2144–2158, 2160–2174	2144
4	0	2176–2190, 2192–2206	2176
	1	2208–2222, 2224–2238	2208
5	0	2240–2254, 2256–2370	2240
	1	2272–2286, 2288–2302	2272
6	0	2304–2318, 2320–2334	2304
	1	2336–2350, 2352–2366	2336
7	0	2368–2382, 2384–2398	2368
	1	2400–2414, 2416–2430	2400
8	0	2432–2446, 2448–2462	2432
	1	2464–2478, 2480–2494	2464
9	0	2496–2510, 2512–2526	2496
	1	2528–2542, 2544–2558	2528
10	0	2560–2574, 2576–2590	2560
	1	2592–2606, 2608–2622	2592
11	0	2624–2638, 2640–2654	2624
	1	2656–2670, 2672–2686	2656
12	0	2688–2702, 2704–2718	2688
	1	2720–2734, 2736–2750	2720
13	0	2752–2766, 2768–2782	2752
	1	2784–2798, 2800–2814	2784
14	0	2816–2830, 2832–2846	2816
	1	2848–2862, 2864–2878	2848
15	0	2880–2894, 2896–2910	2880
	1	2912–2926, 2928–2942	2912
16	0	2944–2958, 2960–2974	2944
	1	2976–2990, 2992–3006	2976
17	0	3008–3022, 3024–3038	3008
	1	3040–3054, 3056–3070	3040
18	0	3072–3086, 3088–3102	3072
	1	3104–3118, 3120–3134	3104

[1] Address 2127, and every 16th address thereafter, is not valid. When two LICs are attached to a CLP, a maximum of 60 attached lines can be simultaneously active. When four LICs are attached to a CLP, up to 100 lines out of the maximum of 120 attached lines can be simultaneously active.

ANS

```
►►──┬─────────────────────────────┬──►◄
    └─ANS=─┬─CONTinue─┬───────────┘
           ├─DELAY────┤
           └─STOP─────┘
```
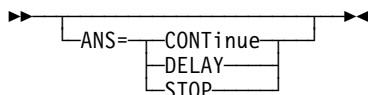
Default: None

Determines session continuation for devices when NCP goes into automatic
network shutdown (ANS) for an owning SSCP because the SSCP is no longer con-
nected to NCP.

CONTinue
Specifies that the session is to continue. ANS=CONTINUE is not valid for BSC
devices.

DELAY
Specifies that ANS processing is to be delayed for this line. This allows cross-
domain resources to remain in session. When an owning SSCP sends an
ACTLINK to the line, ANS processing is performed and the session is termi-
nated. ANS=DELAY is not valid for SDLC devices (see ANS on the PU defi-
nition statement on page 2-369 for more information).

STOP
Specifies that the session is to be terminated.

ANS is not valid for switched subarea lines when you specify DIAL=YES and
PUTYPE=4.

ANSTONE

```
        ┌─ANSTONE=NO──────────┐
►►──────┤                     ├──►◄
        └─ANSTONE=─┬─YES─┬────┘
                   └─NO──┘
```
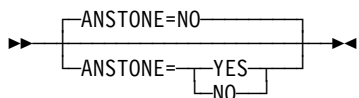
Use: For call-in switched lines in network control mode
Default: NO

Specifies whether NCP transmits an answer tone to the calling device to signify
completion of the line connection.

The program needs to send the tone only when the modem does not send an
answer tone. Most modems in the United States and Canada provide the tone;
therefore, ANSTONE=NO is usually appropriate. If in doubt, consult the supplier or
installer of the modem.

ANSTONE is valid only for incoming calls on switched links.
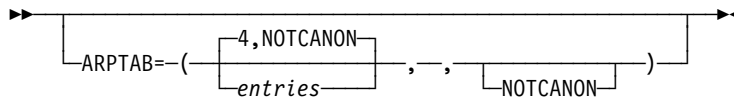
ARPTAB

```
►►─┬────────────────────────────────────┬─►◄
   └─ARPTAB=─(─┬──4,NOTCANON──┬─,─,─┬──────────┬──)─┘
              └─entries─────┘      └─NOTCANON─┘
```

Use: Frame-relay internet devices

For IP over frame relay devices, ARPTAB specifies the maximum number of temporary entries that can exist in the ARP table at any given time.

The ARPTAB keyword must be specified on the LINE definition statement if you plan to dynamically add internet devices to a line without previously defined internet devices.

The value of ARPTAB should not be lower than the number of dynamic resources that are to be added to a frame-relay physical line.

Avoid coding ARPTAB lower than MAXPU since this could prevent NCP from obtaining resources to build an ARP table entry during processing of an Inverse Address Resolution Protocol (InARP) packet received from the adjacent station for a dynamically reconfigured internet host.

entries

> Default: None
> Minimum: 1
> Maximum: 65535
>
> Specifies the maximum number of temporary entries that can exist in the ARP table at any given time. To optimize system operation, code a number equal to the maximum number of internet devices you expect to dynamically add to the frame-relay physical line.

NOTCANON
> Specifies that the hardware addresses received in ARP packets are in noncanonical form.
>
> When you specify ARPTAB=NOTCANON, the hardware address in an ARP packet is interpreted using noninverted format. In this format the *left*most byte of the 2-byte hardware address is transmitted first, and the *left*most bit of each byte is transmitted first.
>
> NOTCANON is the only valid value for frame-relay internet devices.

AUTO

```
►►─┬──────────────────────────────────┬─►◄
   └─AUTO=(─┬─lnbr─┬─┬──────────┬─)─┘
           └─YES──┘ └─,digits──┘
```
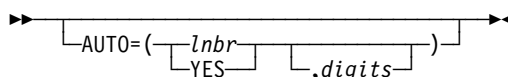
Use: Switched lines (NCP or EP)
Default: None
Example: AUTO=(512,12)

Specifies the relative line number of an ACU and the maximum number of dial digits allowed in a DIAL command. AUTO is required in some cases when there is

no ACU present. An ACU is normally used when a callout operation is performed. When an ACU is installed, you must code the AUTO keyword.

AUTO is valid:

- If you specify DIAL=YES for any switched network control mode or emulation mode line

- If you code VIRTUAL=YES on the GROUP definition statement

- If you do not code AUTOCOPY.

Note: AUTO is not valid if you code AUTOCOPY on this LINE definition statement.

lnbr

Specifies the relative line number for an ACU line. The range of valid line numbers depends on your communication controller model:

Range	*Controller Model*
0–895	3745-x10 and 3745-x1A
0–159	3745-130, 3745-150, 3745-170, 3745-17A
0–31	3745-160

3746 Model 900 SDLC Lines and 3746 Model 900 Switched Frame-Relay Lines: The 3746 Model 900 does not support V.25 auto-call units for SDLC and frame-relay lines, so AUTO=*lnbr* is not valid when you specify a relative line number greater than 2048 for ADDRESS on this LINE definition statement.

YES

Specifies that there is an ACU installed or that a logical ACU is assumed. Code AUTO=YES when:

- A logical ACU is assumed and an ACU is not physically installed.

- You code either LPDA2DL=YES or the YES suboperand on the V25BIS keyword.

VTAM: Specifying AUTO=YES when VIRTUAL=YES is coded provides information to the VTAM initialization process and has no effect on NCP generation.

digits

Use:	EP lines
Default:	See text
Minimum:	0
Maximum:	See text

Specifies the maximum number of dial digits passed on this line by the DIAL command.

If you code the YES suboperand on the V25BIS keyword and include the CRN prefix, the maximum is 64; otherwise, the maximum is 61. For an ACU line, the default is 16; otherwise, the default is 10.

AUTOCOPY

```
►►─────┬──────────────────────────────────────────────────────┬─────►◄
       └─AUTOCOPY=(copies,nextlabel,─┬───D───┬──,──┬───────┬──)─┘
                                     ├───D───┤     └─DONE──┘
                                     └───H───┘
```
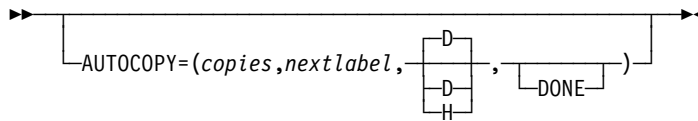
Use: Automatic line generation
Example: AUTOCOPY=(8,A04L001,H)

Specifies the number of copies to be made of a specified line and its devices. It is not necessary to calculate a target address, as required on the AUTOLINE keyword. Instead, code the ENDAUTO definition statement at the end of the statements to be copied.

The following restrictions apply to the AUTOCOPY keyword:

* When you code AUTOCOPY, you must also code NEXT on the CLUSTER, COMP, or TERMINAL definition statement (for BSC and start-stop lines) or on the LU or PU definition statement (for SDLC lines) to control how the labels on subsequent statements are constructed.

  For more information about NEXT, see NEXT on the CLUSTER, LU, PU, or TERMINAL definition statement, or the *NCP, SSP, and EP Resource Definition Guide*.

* When you code AUTOCOPY, you must also specify NEWDEFN=YES on the OPTIONS definition statement.

* When you code an OPTIONS definition statement with AUTOCOPY, the OPTIONS definition statement is not replicated.

* When you specify ADDRESS=NONE, AUTOCOPY is valid only if you specify LEVEL2 and VIRTUAL=NO on the GROUP definition statement.

* AUTOCOPY is not valid under the following conditions:

  – When you are defining a spare or redefinable line

  – When you specify subchannel positions for ADDRESS, ID, or AUTO on the LINE definition statement

  – When you use a physical channel address rather than the logical address (for example, P1 rather than 8)

  – When you specify NPARSC=YES on the GROUP definition statement

  – When you are defining a Token-Ring physical line (ECLTYPE=PHYSICAL), a logical subarea line (ECLTYPE=(LOGICAL,SUBAREA)), or a NTRI frame handler logical line (ECLTYPE=(LOGICAL,FRELAY))

  – When you are defining an Internet Protocol connection (NCST=IP)

  – When you are defining a physical connection to an Ethernet-type LAN (ETHERNET=PHYSICAL)

  – When you are defining a frame-relay physical line (FRELAY=PHYSICAL)

  – When you are defining an ISDN physical line (ISDN=PHYSICAL)

  – When you are defining an ESCON link

  – When you are defining an NRF line (LINEFVT=CXRXFVT)

- You cannot code AUTOCOPY and AUTOLINE on the same line. You can, however, code them on different lines in the same generation definition.

When you code AUTOCOPY you might need to increase the size of your DBWORKFL data set.

Token Ring: AUTOCOPY is valid only for Token-Ring logical peripheral lines. You can use AUTOCOPY rather than AUTOGEN to generate LINE-PU pairs where you can control the label names. Code one LINE definition statement with AUTOCOPY and one PU definition statement with the NEXT keyword, then code the ENDAUTO definition statement. Do not use AUTOCOPY and AUTOGEN in the same group. Do not code UACB if you code AUTOCOPY.

Frame Relay: For frame-relay logical subarea lines, AUTOCOPY increments the value of the data link connection identifier (DLCI) or ADDR on the PU definition statement. For logical peripheral lines, you can use AUTOCOPY rather than AUTOGEN to generate LINE-PU pairs where you can control the label names. For both logical subarea lines and logical peripheral lines, code one LINE definition statement with AUTOCOPY and one PU definition statement with the NEXT keyword, then code the ENDAUTO definition statement. Do not use AUTOCOPY and AUTOGEN in the same group.

ISDN: AUTOCOPY is valid for ISDN B-channels, logical subarea lines, and logical peripheral lines. For ISDN B-channels, AUTOCOPY increments the value of ID for each subsequent LINE statement. You can use AUTOCOPY rather than AUTOGEN to generate LINE-PU pairs where you can control the label names. For both logical subarea lines and logical peripheral lines, code one LINE definition statement with AUTOCOPY and one PU definition statement with the NEXT keyword, then code the ENDAUTO definition statement. Do not use AUTOCOPY and AUTOGEN in the same group.

copies

> Default: None
> Minimum: 1
> Maximum: See text

Specifies the number of copies to make of this line and associated statements. *copies* must be a decimal value. If you code ECLTYPE=(LOGICAL,PERIPHERAL) or FRELAY=(LOGICAL,PERIPHERAL), the maximum number of copies is 5000; otherwise, the maximum number of copies is 128.

Note: If the number of lines to be generated by AUTOCOPY would cause the total number of Token-Ring and frame-relay logical lines to exceed 9999, AUTOCOPY is ignored and no lines are automatically generated.

nextlabel

> Default: None

Specifies the label for the first duplicated line. The label consists of 1 to 8 characters; the first character must be an alphabetical character from A to Z, #, or @. NDF uses this *nextlabel* to determine what positions in the label to use for incrementing label names and the amount of the increment. The value

entered for *nextlabel* must be the same length as the statement label, and must differ from the label in such a way that NDF can identify a numeric field for incrementing. You can use only one field for incrementing. This field can increase in length as it is incremented, but it cannot increase so much that the total label length becomes greater than 8 characters.

The format of the statement *label* is

<p>lo_num_field<s>

The next label generated for copied line definitions is

<p>next_num_field<s>

where:

- *label* is the name specified on this LINE definition statement
- *nextlabel* is the label to be used on the first copy of this LINE definition statement
- *p* is the prefix of the numeric field and is the same for both labels
- *s* is the suffix of the numeric field and is the same for both labels.

NDF subtracts lo_num_field from next_num_field to obtain the value it uses to increment the numeric field for the next label.

The numeric fields cannot differ in length, but must differ in value. next_num_field must be greater than lo_num_field.

If NDF cannot determine how to increment a label, it issues an error message.

D|H

Default: D

Specifies whether the increment and the field to be incremented are decimal (D) or hexadecimal (H). This value does not need to match the value you code for the NEXT keyword on the CLUSTER, COMP, LU, PU, or TERMINAL definition statement.

DONE
NDF adds DONE to the AUTOCOPY keyword in the NEWDEFN file to indicate that the line has been processed. For example, if you specify AUTOCOPY=(8,A04L001) NDF writes AUTOCOPY=(8,A04L001,DONE) to the NEWDEFN file. You can code DONE to cause NDF to ignore this AUTOCOPY keyword.

AUTODL

```
         ┌─AUTODL=YES─────────┐
►►───────┤                    ├──►◄
         └─AUTODL=─┬─YES─┬─────┘
                   └─NO──┘
```
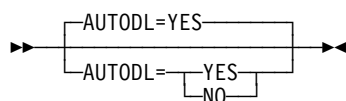
Use: Switched X.21 lines
Default: YES

Specifies whether the X.21 data circuit-terminating equipment (DCE) accepts dial digits.

AUTODL=YES causes NCP to run the normal callout procedure for this line. For a DCE that does not accept dial digits, code AUTODL=NO.

AUTOLINE

```
►►──────┬──────────────────────────┬────►◄
        └─AUTOLINE=(target────────)─┘
                          └─,DONE─┘
```
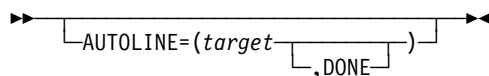
Use: Automatic line generation
Default: None
Example: AUTOLINE=40

Causes NDF to create copies of this LINE definition statement and all definition statements that are subordinate to it, and specifies a target that NDF will work toward as it increments the address of each copy. For more information on the autoline function, see the *NCP, SSP, and EP Resource Definition Guide*.

When you code AUTOLINE, you must also code the ENDAUTO definition statement and NEWDEFN=YES on the OPTIONS definition statement.

target

Default: None

Specifies a decimal number that NDF will work toward as it increments the address of each copied line; in this way, *target* determines how many copies to make.

DONE

NDF adds DONE to the AUTOLINE keyword in the NEWDEFN file to indicate that the line has been processed. For example, if you specify AUTOLINE=40, NDF writes AUTOLINE=(40,DONE) to the NEWDEFN file. You can code DONE to cause NDF to ignore this AUTOLINE keyword.

AUTOLINE is not valid under the following conditions:

- When you are defining a spare or redefinable line
- When you specify subchannel positions for ADDRESS, ID, or AUTO on the LINE definition statement
- When you code ADDRESS=NONE on the line definition statement
- When you specify NPARSC=YES on the GROUP definition statement
- When you are defining a Token-Ring line (ECLTYPE)
- When you are defining an Internet Protocol connection (NCST=IP)
- When you are defining a connection to an Ethernet-type LAN (ETHERNET)
- When you are defining a frame-relay line (FRELAY)
- When you are defining an ESCON link

For the AUTOLINE function to work correctly, you must embed the numeric value coded for the first suboperand of ADDRESS in the label of this LINE definition statement. You must also embed this value in the labels of all the statements that are subordinate to this LINE definition statement.

You cannot code AUTOLINE and AUTOCOPY on the same line. You can, however, code them on different lines in the same generation definition. See AUTOCOPY on the LINE definition statement on page 2-196 for more information.

AUTUACB
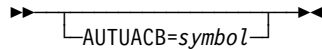
```
►►──┬─────────────────────┬──►◄
    └─AUTUACB=symbol─┘
```

Use:	User-written line control
Default:	None

Specifies the name of the user adapter control block associated with the relative line number specified by AUTO on this LINE definition statement. AUTUACB is required if user-defined line code is used to support a switched line using the auto-call facility. If the auto-call facility or user-defined line code is not supported, AUTUACB is not valid.

AVGPB

```
►►──┬────────────────────┬──►◄
    └─AVGPB=bytes─┘
```
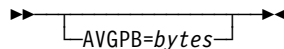
Use:	BSC and start-stop devices in network control mode
Default:	Value of BFRS on the BUILD definition statement
Minimum:	1
Maximum:	65535

Specifies the average number of bytes expected from a BSC or start-stop device when it is polled. NCP uses this byte count to determine whether there is enough buffer space to receive the expected data from a device that answers its poll.

PU is the lowest level definition statement on which you can specify AVGPB for an SDLC line. See AVGPB on the PU definition statement on page 2-372 for more information.

BALANCE

```
          ┌─BALANCE=(0,D)──────────────────┐
►►──┼───────────────────────────────────┼──►◄
          │             ┌─0─┐       ┌─D─┐  │
          └─BALANCE=(──┼───┼──,──┼─D─┼──)─┘
                        └─factor─┘  ├─D─┤
                                    └─H─┘
```
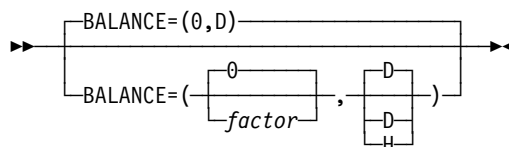
Use:	NTRI physical lines
Default:	0,D
Minimum:	0,D
Maximum:	16,D or 10,H

Specifies the balancing factor to be used by a NTRI physical line. Specify a balancing factor when you want a physical line with a duplicate locally administered MAC address to participate in the balancing of all peripheral connections (established through incoming call requests) across all associated duplicate TICs. For more information, refer to *Duplicate TIC Addressing* in the *NCP, SSP, and EP Resource Definition Guide.*

Do not specify a balancing factor when the NTRI physical line has a unique locally administered MAC address.

BALANCE definition formats and definition rules are as follows:

* BALANCE=(*factor*,D) — D means that BALANCE is specified in decimal.
* BALANCE=(*factor*,H) — H means that BALANCE is specified in hexadecimal.

Decimal is assumed if you do not specify a value.

BEACTO

```
                ┌─BEACTO=52───────┐
►►──────────────┤                 ├──►◄
                └─BEACTO=seconds──┘
```
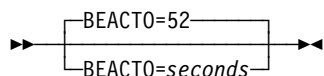
Use: Token-Ring resources
Default: 52
Minimum: 52
Maximum: 600

Specifies, in seconds, how long NCP is to allow a Token Ring to beacon before considering it permanently down. BEACTO is valid only if you specify ECLTYPE=PHYSICAL on the GROUP definition statement.

3746 Model 900 Token Ring: If you specify ADDRESS=2080, NCP uses BEACTO=52 regardless of the value you specify.

BUFETTE

```
                ┌─BUFETTE=2───────┐
►►──────────────┤                 ├──►◄
                └─BUFETTE=buffers─┘
```
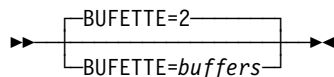
Use: Start-stop lines in emulation mode
Default: 2
Minimum: 2
Maximum: 255

Specifies how many 4-byte buffers are to be allocated to this line.

These buffers are used collectively as a data storage buffer to prevent overrun of data received from the line when the channel is in a slowdown or lockout condition.

BUFSIZE

```
                ┌─BUFSIZE=64────┐
►►──────────────┤               ├──►◄
                └─BUFSIZE=size──┘
```
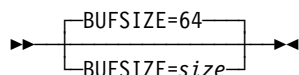
Use: BSC lines in emulation mode
Default: 64
Minimum: 4
Maximum: 254

Specifies the size, in bytes, of the two cycle-steal buffers that are generated for each line in normal mode (CSPMODE=NORMAL). BUFSIZE is valid only when you specify CSPMODE=NORMAL.

size must be specified in even increments.

CA

```
►►──────────────────────────►◄
      └─CA=──┬─TYPE6─────┬──┘
             ├─TYPE6-TPS─┤
             ├─TYPE7─────┤
             └─TYPE7-TPS─┘
```
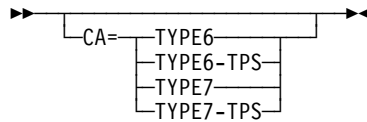
| Use: Channel links (NCP or EP)
Default: See text

Specifies the type of channel adapter installed in the communication controller in which NCP resides. If the channel adapter has a two-processor switch, code TYPE6-TPS or TYPE7-TPS. If you use a two-processor channel adapter, code an even number on the ADDRESS keyword if you are specifying a logical address, or an odd number if you are specifying a physical position.

Code CA=TYPE6 or CA=TYPE6-TPS for a type 6 channel adapter. For migration purposes, you can specify CA=TYPE7 or CA=TYPE7-TPS for a type 6 channel adapter. When you do this, it will function as a type 6 channel adapter.

Code CA=TYPE7 or CA=TYPE7-TPS for a type 7 channel adapter. For migration purposes, you can specify CA=TYPE6 or CA=TYPE6-TPS for a type 7 channel adapter. When you do this, it will function in compatibility (non-buffer-chaining) mode, that is, as a type 6 channel adapter.

For a channel adapter associated with lines operating in emulation mode, TYPE6 is the default; otherwise, TYPE7 is the default.

ESCON Links: The CA keyword is not valid for ESCON links.

CALINE

```
►►──────────────────────────►◄
      └─CALINE=name─┘
```
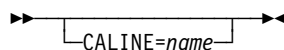
Use: Lines attached to an IBM modem with the data multiplexer feature
Default: None

Specifies the name of the link attached to channel A of the modem.

Code CALINE for this line only if you specify LPDATS=LPDA1 on this LINE definition statement.

Specify CALINE only if the line you define is *not* attached to channel A, but *is* attached to channel B, C, or D. Channel A is the LPDA control channel. If you specify CHANLA=NO (CHNLZ=YES), code CALINE.

For casual connection, CALINE is not valid if you specify MODE=SECONDARY on the GROUP definition statement.

CALL

```
          ┌─CALL=IN─────────┐
►►────────┴─────────────────┴──►◄
          └─CALL=──┬─IN────┬─
                   ├─OUT───┤
                   └─INOUT─┘
```
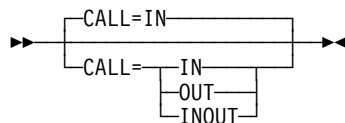
Use: BSC, ISDN B-channel, or start-stop switched lines in network control
            mode

Default: IN

Specifies whether this line is to be used for incoming calls, outgoing calls, or both.
CALL is valid only if you specify DIAL=YES on the GROUP definition statement.

IN

Specifies that this line is to be used only for incoming calls (stations call the
controller).

OUT

Specifies that this line is to be used only for outgoing calls (the controller calls
stations). If you code CALL=OUT, also code the DIALSET keyword to define
the name of the dial set for this line.

INOUT

Specifies that this line is to be used for both incoming and outgoing calls. If
you code CALL=INOUT, also code the DIALSET keyword to define the name of
the dial set for this line.

VTAM: For lines codes with LNCTL=SDLC, CALL provides information only to the
         VTAM initialization process and has no effect on the generation of NCP.
         See the *VTAM Resource Definition Reference* for information on this
         keyword.

CASDL

```
►►──────────────────────────►◄
     └─CASDL=seconds─┘
```
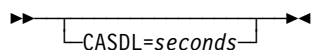
Use: Channel links

Default: Value of TIMEOUT on the LINE definition statement

Minimum: 0.0

Maximum: 840.0

Defines the channel adapter slowdown limit, that is, how long this channel link can
block inbound data before signaling that this station is inoperative.

CASDL is not valid if you specify TIMEOUT=NONE on this LINE definition state-
ment.

ESCON Links: The CASDL keyword is valid for logical lines only; do not code
             CASDL for ESCON physical lines.

CCLID

```
                 ┌─CCLID=NO────────┐
►►───────────────┴─────────────────┴──►◄
          └─CCLID=──┬─YES─┬─┘
                    └─NO──┘
```
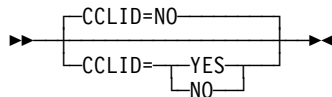
Use: Switched X.21 lines
Default: NO

Specifies that NCP is to support called/calling line ID (CCLID). With NCP V7R3 or earlier, to fully exploit the CCLID function, the X.21 SH/MPS program product is required. Starting with NCP V7R4, you can fully exploit the CCLID function with NCP when the X.21 SH/MPS program product is not available.

CHANLA

```
►►──┬──────────────────────┬──►◄
    └─CHANLA=──┬─YES─┬──────┘
               └─NO──┘
```
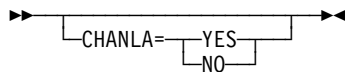
Use: Lines attached to an IBM modem with a data multiplexer
Default: See text

Specifies whether this line is attached to the first-level link LPDA control channel (channel A) on the IBM modem. Each IBM modem with the data multiplexer feature has only one LPDA control channel (channel A).

CHANLA has no default if you do not code PORT and if you code LPDATS=LPDA2 and CHNLZ=YES. If you code CHANLA=NO or omit CHANLA, you must specify the PORT keyword. If you code CHANLA=YES, PORT may be omitted.

For casual connection, CHANLA is not valid if you specify MODE=SECONDARY on the GROUP definition statement.

Table 2-59 shows the effect on CHANLA when you code certain keywords on the GROUP and LINE definition statements.

Table 2-59. NDF Values for CHANLA on the LINE Definition Statement

When You Code This	The Effect on CHANLA Is
LINE: LPDATS=LPDA2, CHNLZ=YES, and PORT=A	The default is CHANLA=YES.
LINE: LPDATS=LPDA2 and CHNLZ=YES and PORT=B, PORT=C, or PORT=D	The default is CHANLA=NO.
GROUP: ECLTYPE	The only valid value is CHANLA=NO.
GROUP: ETHERNET=PHYSICAL	The only valid value is CHANLA=NO.
GROUP: FRELAY	The only valid value is CHANLA=NO.

CHECK

```
              ┌─CHECK=NODCD──────────┐
►►────────────┤                      ├────►◄
              └─CHECK=──┬─DCD───┬─────┘
                        └─NODCD─┘
```
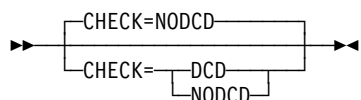
| Use: Switched, duplex start-stop lines in emulation mode
Default: NODCD

| Specifies whether the control program (NCP or EP) is to use data-carrier-detect for the line represented by this LINE definition statement. Use of this option prevents access to an application program's data by a device that dials the controller over this line at the moment the existing connection to a different device is lost. Continuous monitoring of the data-carrier-detect signal from the modem gives positive assurance that the switched line connection still exists.

CHECK is valid only if DUPLEX=FULL is coded on the LINE definition statement and LNCTL=SS and DIAL=YES are coded on the GROUP definition statement.
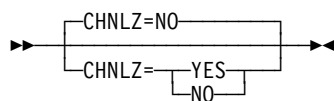
CHNLZ

```
              ┌─CHNLZ=NO─────────┐
►►────────────┤                  ├────►◄
              └─CHNLZ=──┬─YES─┬───┘
                        └─NO──┘
```

Default: NO

Specifies whether this line is attached to an IBM modem with the data multiplexer feature.

If you specify CHNLZ=YES, also specify CORNUM and either CALINE or CLINES, depending on whether or not this line is specified as channel A. If you specify CHNLZ=NO, also specify CHANLA=NO and omit CORNUM, CALINE, and CLINES. You cannot specify CHNLZ=YES if you define LPDATS=LPDA2 on this LINE definition statement.

For casual connection, CHNLZ is not valid if you specify MODE=SECONDARY on the GROUP definition statement.

CHNLZ is not valid when you code PUTYPE=4.

For IBM 786x modems extended LPDA2 capability, specify CHNLZ for lines with LPDA1, LPDA2, or no LPDA mode, regardless of which port on the modem the line is connected to. You may code CHNLZ=YES if you specify LPDATS=NO.

Table 2-60 shows the effect on CHNLZ when you code certain keywords on the GROUP definition statement.

Table 2-60. NDF Values for CHNLZ on the LINE Definition Statement

When You Code This on GROUP	The Effect on CHNLZ Is
ETHERNET=PHYSICAL	The only valid value is CHNLZ=NO.
FRELAY	The only valid value is CHNLZ=NO.

CHNPRI

```
                  ┌─CHNPRI=NORMAL─────┐
►►────────────────┼───────────────────┼──►◄
                  └─CHNPRI=─┬─NORMAL─┬─┘
                           └─HIGH───┘
```
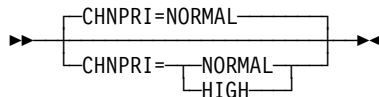
Use: BSC lines in emulation mode
Default: NORMAL

Specifies the priority of the emulation subchannel associated with this line relative to the other emulation subchannels. CHNPRI is valid only for a BSC line in emulation mode.

CHNPRI=NORMAL is ordinarily appropriate unless the data rate of the line is 19200 bps or more and the majority of the remaining lines are slow-speed lines (2400 bps or less).

CIR

```
          ┌─CIR=NO──────────────────────────────────────────┐
►►────────┼─────────────────────────────────────────────────┼──►◄
          └─CIR=─┬─(YES,─┬────────┬─,─┬──────┬──┬────────┬─)─┘
                 │       │  ┌─7─┐ │   │ ┌─25─┐│  └─period─┘
                 │       └─precision─┘   └─mir%─┘
                 └─NO─────────────────────────────────────────
```
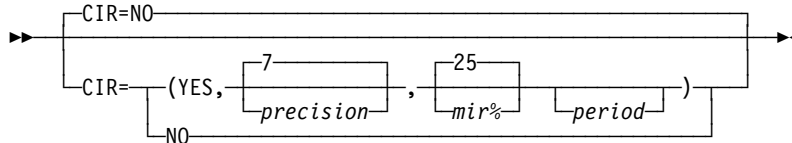
Use   3746 Model 900 attached frame-relay physical lines
      3746 Model 900 attached ISDN physical lines
Default: NO

Specifies whether the Committed Information Rate (CIR) algorithms should be used for bandwidth management.

CIR is the information transfer rate to which the network is committed under normal conditions. For more information about CIR on 3746 Model 900 attached frame-relay physical lines, see CIR on the DLCI definition statement on page 2-85.

If CIR=YES is coded but no DLCI statements are coded, then the 3746 Model 900's automatically adjusting CIR algorithms are used.

precision

Default: 7
Minimum: 6
Maximum: 10

Specifies an integer used by the CIR adaptive algorithm. Precision determines the speed with which the CIR adaptive algorithm reacts to a changing environment. The larger the value, the slower the reaction. A large value would suit an environment with stable network conditions, and a small value would suit an environment with changing network conditions.

mir%

Default: 25
Minimum: 1

| Maximum: 100

| Specifies an integer used by the CIR adaptive algorithm to determine the minimum information rate that the algorithm can reach for a DLCI.

| *period*

| Default: Set by the 3746 Model 900 based on *precision*
| Minimum: .1
| Maximum: 10.0

| Specifies, in seconds, the interval used by the CIR adaptive algorithm to determine the CIR. Valid values are in intervals of a tenth of a second.

| See the *3745 Communication Controller Models A, 3746 Expansion Unit Model 900: Migration and Planning Guide*, GA33-0183, for more information about the Model 900 CIR processing.

CLINES

```
►►─┬──────────────────────────────────────────────────┬─►◄
   └─CLINES=(─┬──────────────┬─,─┬──────────────┬─,─┬──────────────┬─)─┘
             └─chan_B_name─┘     └─chan_C_name─┘     └─chan_D_name─┘
```

Use: Lines attached to an IBM modem with a data multiplexer
Default: None
Example: CLINES=(,NT5L003,NT5L004)

Specifies the names of the lines attached to channels B, C, and D of the IBM modem to which this line is attached. CLINES is valid only if (1) this line is attached to channel A (the LPDA control channel), and (2) you code LPDATS=LPDA1 on this LINE definition statement.

If no line is attached to a specific channel, omit the suboperand for that channel, but include the comma to represent the omitted suboperand. Omit CLINES if no other lines are attached to the modem.

Casual Connection: CLINES is not valid if you code MODE=SECONDARY on the GROUP definition statement.

| CLLM

```
      ┌─CLLM=NO──────────────────┐
► ──┼──────────────────────────┼─►◄
      │              ┌─11─┐      │
      └─CLLM=─┬─(YES,─┼────┼─)─┘
             │       └─Ty─┘
             └─NO─────────────
```

| Use: 3746 Model 900 attached frame-relay physical lines
| Default: NO

| Specifies whether Consolidated Link Layer Management (CLLM) will be used. CLLM=YES is valid only when CIR=YES is coded.

| *Ty*

| Default: 11
| Minimum: 5
| Maximum: 30

| Specifies, in seconds, the Ty timer used to determine when the congestion or
| outage condition has cleared. Valid values are in intervals of one second.

| The Ty timer is a complement of the Tx timer, run by the DCE, which resends
| CLLM congestion and outage messages every Tx seconds as long as the con-
| dition continues.

| See the *3745 Communication Controller Models A, 3746 Expansion Unit Model
| 900: Migration and Planning Guide*, GA33-0183, for more information about the
| Model 900 CLLM processing.

CLOCKNG

```
►►──┬───────────────────────────┬──►◄
    └─CLOCKNG=─┬─INT──────────┬─┘
              ├─DIRECT────────┤
              └─(EXT─┬──────┬─)┘
                     └─,LIC─┘
```
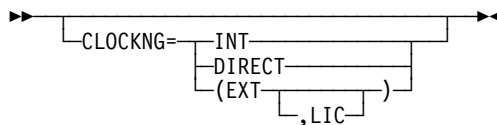
| Use: NCP or EP
Default: See text

Specifies whether the modem (data set) or the communication scanner for the line
provides clocking.

Notes:

1. Be sure to spell CLOCKNG without the letter *I*.

2. For more information, see SPEED on the LINE definition statement on
   page 2-276.

If your line is directly attached to another controller, you must code
CLOCKNG=DIRECT in the NCP generation definition for the controller that pro-
vides the clocking. In the generation definition for the other controller, code
CLOCKNG=EXT.

If your line is attached to a modem that provides clocking, code CLOCKNG=EXT.

Note the following special coding situations for CLOCKNG:

- If this LINE definition statement represents a BSC or SDLC line, the default is
  CLOCKNG=EXT.

- If the LINE definition statement represents a start-stop line, the default is
  CLOCKNG=INT.

- If you define a multiple-terminal-access line, do not code CLOCKNG.

- If the line is an X.21 line, code CLOCKNG=EXT.

Frame Relay: For frame-relay physical resources, CLOCKNG=INT is not valid.
The default is CLOCKNG=EXT.

ISDN: For ISDN physical resources, CLOCKNG=INT is not valid. The default is
CLOCKNG=EXT.

INT
Specifies that the scanner provides clocking, but does not propagate it to the
attached data terminal equipment (DTE). The scanner synchronizes the
received data signal; a different clock is used at each end of a link. You may
use either a directly attached cable (no modem) or a modem cable. The
SPEED keyword is required.

DIRECT
Specifies that the scanner provides clocking and propagates it to the DTE. The
data signal received by the controller is synchronized with the controller's clock.
You must use a directly attached cable. The SPEED keyword is required.

For more information about clocking, see the "Direct Attachment of Data Ter-
minal Equipment" section of the *NCP and EP Reference*.

EXT
Specifies that the modem or other external source provides clocking. You can
use either a directly attached cable or a modem cable. If you specify
CLOCKNG=EXT, you need to code the SPEED keyword when:

- You use NPM
- You access an X.21 network (X21NETWK=OLD or 1984 on the GROUP
  definition statement)
- Your line is a 3746 Model 900 SDLC line
- Your line is a 3746 Model 900 frame-relay line
- Your line is an SDLC peripheral or a frame-relay physical peripheral line
  that supports APPN HPR data

If you specify a speed of 2400 bps or more, clocking is provided on the
external cable. If you specify a speed of 1200 bps or less, clocking will not
appear on the external cable.

(EXT,LIC)
Specifies that the modem or other external source provides clocking and that
the line is attached to an X.21 interface (LIC type 4). Specify
CLOCKNG=(EXT,LIC) when you are using an X.21 interface but are *not*
accessing the X.21 network (X21NTWK=NO on the GROUP definition state-
ment). The SPEED keyword is required.

Note: To calculate the LIC weight for a low-speed line, see Appendix B, "Line
Weights for Low-Speed Scanners," in the *Connection and Integration Guide*
for your communication controller model.

CODE

```
                    ►►──┬────────────┬────────►◄
                        └─CODE=──┬─ASCII───────┬
                                 ├─BCD─────────┤
                                 ├─BCD2────────┤
                                 ├─COR─────────┤
                                 ├─COR2────────┤
                                 ├─DIC1────────┤
                                 ├─DIC3────────┤
                                 ├─EBCD────────┤
                                 ├─EBCDIC──────┤
                                 ├─ITA2────────┤
                                 ├─KATAKANA────┤
                                 ├─USASCII─────┤
                                 ├─USASCII-T───┤
                                 └─ZSC3────────┘
```
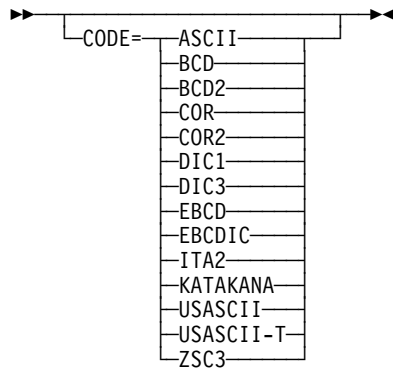
Use: Start-stop and BSC lines (NCP or EP)
Default: None

Specifies the transmission code used to communicate with devices attached to this line. Ask the system designer which code is used. The "Line Character Codes" section of *NCP and EP Reference Summary and Data Areas*, Volume 2, contains the translation tables associated with each transmission code.

EP Standalone: CODE is only for BSC lines. Two parameters can be coded, CODE=(*code1,code2*), if FEATURE=(DUALCODE) is specified for the line. *code1* and *code2* can each be specified as EBCDIC, USASCII, or USASCII-T. If USASCII is specified, LRC checking is performed for all data transmitted and received over the line. If USASCII-T (transparent USASCII) is specified, CRC checking is performed for all data (transparent and nontransparent) transmitted and received over the line.

If FEATURE=(DUALCODE) is specified and CODE is omitted, *code1* defaults to EBCDIC. If FEATURE=(DUALCODE) is specified and the *code2* parameter is omitted, the default is one of the following:

- USASCII if *code1* is EBCDIC or if the CODE keyword is omitted
- EBCDIC if *code1* is USASCII or USASCII-T

USASCII-T is valid only if the line represented by the LINE definition statement is serviced by a scanner in normal mode.

Table 2-61 on page 2-211 lists the codes you can define for each type of device. (The WTTY column refers to various European teletypewriters using either the ITA2 or ZSC3 transmission codes.)

CODE can be specified only for the devices shown in Table 2-61. If you specify a code not listed, it is ignored. Also, omit CODE if the line is a multiple-terminal-access line or if the LINE definition statement is for an SDLC link.

Table   2-61. Values for CODE on the LINE Definition Statement

Transmission Code	CODE=	Type of Device					
		BSC Devices	IBM 1050	IBM 2740	IBM 2741	TWX	WTTY
ASCII	ASCII					•	
Binary Coded Decimal 1	BCD		•	•	•		
Binary Coded Decimal 2	BCD2		•		•		
Correspondence 1	COR			•	D		
Correspondence 2	COR2				•		
Data Interchange Code 1	DIC1					D	
Data Interchange Code 3	DIC3					•	
Extended Binary Coded Decimal	EBCD		D	D	•		
Extended BCD Interchange Code	EBCDIC	D					
International Telegraph Alphabet No. 2	ITA2						D
Katakana	KATAKANA		•	•			
USA Standard Code for Information Inter-change	USASCII	•					
USASCII (transparent)	USASCII-T	EP					
Figure Protected Code	ZSC3						•

•   May be coded
D   Default value
EP   Valid only in emulation mode

COMRATE
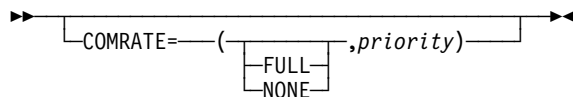
Use: Frame-relay logical lines (FRTEs)
             NTRI frame handler logical lines
             ISDN logical subarea and peripheral lines
Default: COMRATE=(FULL,1) for FRTEs and for ISDN logical lines
             COMRATE=(,1) for NTRI frame handler logical lines

For FRTEs and ISDN logical lines, COMRATE specifies whether certain frames
sent over this line can be discarded by the frame-relay network to expedite the
processing of other frames when the network is congested. For FRTEs, ISDN
logical lines, and NTRI frame handler logical lines, COMRATE also specifies the

relative transmission priority for a terminating equipment subport (TESP) or logical link.

FULL│NONE

Valid on GROUP and LINE definition statements only.

Use: Frame-relay TESPs
          ISDN logical subarea and peripheral lines
Default: FULL

Frames in a frame-relay network can be designated as *discard-eligible*. When the network is congested, discard-eligible frames can be discarded to facilitate the processing of frames that are not discard-eligible. Designating frames as not discard-eligible helps ensure that they are transmitted successfully through the network.

FULL specifies that none of the frames sent by or for this TESP will be marked discard-eligible.

NONE specifies that all frames sent by or for this TESP will be marked discard-eligible except for the following types:

- S-frames (RR, RNR, and REJ), including RRs piggybacked on I-frames
- Network priority frames (VR pacing and TG sweep)
- XIDs

ISDN: In a pure ISDN environment, this suboperand does not have any effect on throughput because the ISDN provides dedicated bandwidth for the B-channel connections. However, if part of the connection crosses a frame-relay network, for example one of the DTEs is an FRTE, the value that you code for this suboperand might affect which frames the frame-relay network discards when congestion occurs.

priority

For more information see COMRATE on the PU definition statement on page 2-375.

Use: Frame-relay logical stations (FRTEs)
           NTRI frame handler logical lines
           ISDN logical subarea and peripheral lines
Minimum: 1
Maximum: 64
Default: 1

Specifies the relative transmission priority of a TESP or logical line. The value of the *priority* suboperand is multiplied by the value of the DATABLK keyword on the physical LINE definition statement (*priority*∗DATABLK) to determine the guaranteed minimum number of bytes that a resource can transmit before a different resource must be given a transmit opportunity.

CONFIG

```
          ┌─CONFIG=NONSW─┐
►►────────┤              ├────────►◄
          └─CONFIG=─┬─SW───┬─┘
                    └─NONSW─┘
```
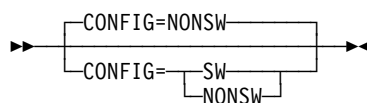
Use: Subarea links
Default: NONSW

Specifies whether the SDLC subarea link between two communication controllers is switched (CONFIG=SW) or nonswitched (CONFIG=NONSW).

If the path is switched, NCP uses switched line procedures to activate and deactivate the line. These procedures can include monitoring the line for ring indicator signals (if RING=YES) and presenting an answer tone (if ANSTONE=YES) when this communication controller is called by the other communication controller.

CONFIG=SW is valid only if you code LNCTL=SDLC and DIAL=NO on the GROUP definition statement associated with this line and PUTYPE=4 on a subordinate PU definition statement.

CONFIG is not valid for multipoint subarea links.

CORNUM

```
►►──┬─────────────────────┬──►◄
    └─CORNUM=number─┘
```
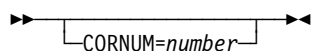
Use: Lines attached to an IBM modem with a data multiplexer
Default: None
Minimum: 1
Maximum: 40

Specifies a 2-byte correlation number for this line. Code the same value for each line attached to the same modem.

If you code CHNLZ=YES, specify CORNUM. If you specify CHNLZ=NO, omit CORNUM.

All LINE definition statements coded with the same correlation number must have the same LPDA mode of operation. In addition, they must (1) all be attached to TSS or HPTSS adapters, or (2) all be attached to the 3746 Model 900.

A maximum of four lines can have the same correlation number.

For casual connection, CORNUM is not valid if you specify MODE=SECONDARY on the GROUP definition statement.

For IBM 786x modems extended LPDA2 capability, CORNUM is required if you code CHNLZ=YES and LPDATS=LPDA2. All lines connected to the same DMPX modem must have the same CORNUM. You may code either LPDATS=LPDA2 or LPDATS=NO for lines with the same CORNUM.

CRATEDLY

```
►►─────────────────►◄
        └─CRATEDLY=n─┘
```

Use: 3746 Model 900 frame-relay/ISDN physical line congestion control

Default: Set by the Model 900 based on line speed
Minimum: 0.1
Maximum: 25.4

The value of **n** specifies the CRATEDLY threshold in seconds. Valid values are in intervals of a tenth of a second.

When a frame is removed from the 3746 Model 900 software queue to be put on the 3746 Model 900 hardware queue for transmission and COMRATE has been exceeded, the Forward Explicit Congestion (FECN) indicator is set on if the frame has been on the software queue for less than the amount of time specified by CRATEDLY.

If the frame has been on the software queue in excess of the amount of time specified by CRATEDLY but less than the amount of time specified by DSCRDDLY, then the following action will be taken:

- If more than 100% of the station's COMRATE has been used, the frame is discarded.

- If more than 75% but less than or equal to 100% of the station's COMRATE has been used, the frame is transmitted with the FECN indicator set to B'1'.

- If less than 75% of the station's COMRATE has been used, the frame is transmitted.

See the *3745 Communication Controller Models A, 3746 Expansion Unit Model 900: Migration and Planning Guide*, GA33-0183, for more information about the Model 900 congestion control processing.

Note: CRATEDLY must be less than or equal to DSCRDDLY.

CRRATE
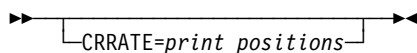
```
►►─────────────────────────────►◄
        └─CRRATE=print_positions─┘
```

Use: Start-stop lines in network control mode
Default: None
Minimum: 1
Maximum: 255

Specifies how many print positions the carriage returns for each idle character NCP sends to a printer on this line.

If you omit CRRATE, 40 print positions for each idle character is the default for TWX teletypewriter terminals. Thirteen print positions is the default for IBM 1050 terminals equipped with the accelerated carrier return feature. Ten positions per idle character is the default for all other terminals.

CRRATE is valid only if you code LNCTL=SS on the GROUP definition statement and your lines operate in network control mode.

CSPMODE

```
►►─────┬─────────────────────────────┬─►◄
        └─CSPMODE=─┬─NOrmal──┬────────┘
                    └─EMulate─┘
```
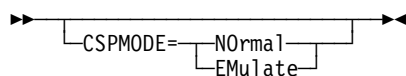
Use: Start-stop and BSC lines in emulation mode
Default: See text

Specifies the mode of operation for emulation mode lines. Code CSPMODE=EMULATE for start-stop lines. CSPMODE=EMULATE is the default for start-stop lines and lines supported by user-defined code. CSPMODE=NORMAL is the default for BSC lines.

CU

```
                  ┌─CU=2703─────────┐
►►────────────────┼─────────────────┼──►◄
                  └─CU=─┬─2701───┬───┘
                        ├─2702───┤
                        ├─2703───┤
                        └─IR2703─┘
```
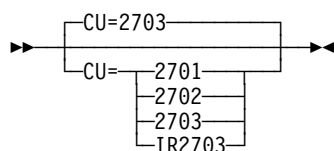
Use: Emulation mode
Default: 2703

| Specifies which transmission control unit functions NCP or EP is to emulate for this line.

| CU=IR2703 allows you to select the sense code that NCP or EP returns when a WRITE command is issued to a line that is not enabled. If you code CU=IR2703, | NCP or EP issues a sense code of IR (intervention required); otherwise, NCP or | EP issues a sense code of command reject (CR).

CUTOFF

```
                  ┌─CUTOFF=NO───────┐
►►────────────────┼─────────────────┼──►◄
                  └─CUTOFF=─┬─subblocks─┬─┘
                            └─NO────────┘
```
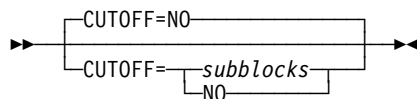
Use: Start-stop and BSC lines in network control mode
Default: NO
Minimum: 1
Maximum: 255

Specifies the maximum number of subblocks NCP is to accept from a device attached to this line. (A subblock is the sequence of message text occupying the number of buffers specified by TRANSFR on this definition statement.) If NCP

receives this number of subblocks before receiving an end-of-block character from the device, it breaks off the transmission.

NCP continues to accept message text from the device until it receives an end-of-block character.

For BSC 3270 lines, code CUTOFF=1.

CUTYPE

```
            ┌─CUTYPE=3271─────────┐
►►──────────┤                     ├────────►◄
            └─CUTYPE=──┬─3271─┬───┘
                       ├─3275─┤
                       ├─2845─┤
                       └─2848─┘
```
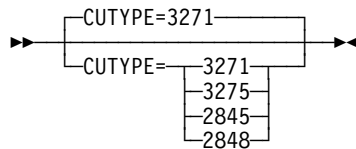
| Use: Start-stop and BSC lines (NCP or EP)
Default: 3271

Specifies the type of control unit to which the individual devices in a clustered station are attached. If the line defined by this definition statement operates in network control mode, specify the control unit in CUTYPE on the CLUSTER definition statement and omit CUTYPE from the LINE. If the line operates both in network control mode and in emulation mode, specify CUTYPE on the GROUP definition statement only, the LINE definition statement only, both the GROUP and the CLUSTER definition statements, or both the LINE and the CLUSTER definition

| statements. If the line operates only in emulation mode, specify CUTYPE only on
| the GROUP definition statement or only on the LINE definition statement.

CUTYPE=3271 or CUTYPE=3275 is valid only when LNCTL=BSC is coded. CUTYPE=2845 or CUTYPE=2848 is valid only for EP and only if LNCTL=SS on the GROUP definition statement for this line.

If more than one control unit type is attached to this line, specify either type on CUTYPE. For example, if both a 3271 and a 3275 are attached to the line, code either CUTYPE=3271 or CUTYPE=3275.

DATABLK

```
                 ┌─DATABLK=2048─┐
►►───────┼──────────────┼───────►◄
                 └─DATABLK=bytes─┘
```
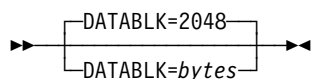
Use: Frame-relay, ISDN, and Token-Ring physical lines
Minimum: 265
Maximum: 16732
Default: 2048

Specifies the minimum guaranteed transmission bandwidth in bytes that can be allocated to a resource or to a protocol. The DATABLK keyword is used with the *priority* suboperand of the COMRATE keyword or FRRATES keyword to determine the relative priority among resources or among supported protocols. The value of the *priority* suboperand is multiplied by the value of the DATABLK keyword (*priority*∗DATABLK) to determine the guaranteed minimum number of bytes that a resource or protocol can transmit before a different resource or protocol must be given a transmission opportunity.

DATRATE

```
                 ┌─DATRATE=LOW──────┐
►►───────┼──────────────────┼───────►◄
                 └─DATRATE=─┬─LOW──┬─┘
                           └─HIGH─┘
```
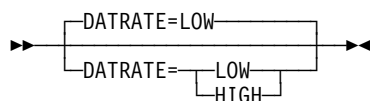
Use: NCP or EP
Default: LOW

Specifies the data rate NCP is to use for a link attached to a dual-rate modem. Code DATRATE=HIGH to use the higher rate. Code DATRATE=LOW (or omit the keyword) to use the lower rate.

Do not code DATRATE if the modem has only one data rate.

DELAY

```
                 ┌─DELAY=0.1─────┐
►►───────┼───────────────┼───────►◄
                 └─DELAY=seconds─┘
```
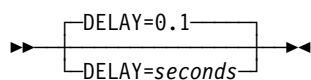
Use: Channel links
Default: 0.1
Minimum: 0
Maximum: See text

Specifies how long NCP can delay from the time data is available to send to the host to the time NCP presents an attention signal to the channel. You can specify this value as an integer or with one decimal place.

The maximum value for DELAY is half of the value coded on TIMEOUT. If you specify TIMEOUT=NONE, the maximum value is 420.0.

ESCON Links: The DELAY keyword is valid for logical lines only; do not code DELAY for ESCON physical lines.

DIALALT

```
                  ┌─DIALALT=NONE─┐
►►────────────────┴──────────────┴────►◄
       └─DIALALT=─┬─name─┬─┘
                  └─NONE─┘
```
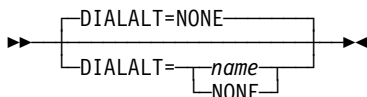
Use: Nonswitched point-to-point and BSC multipoint lines
Default: NONE

Specifies the name of the dial set (DIALSET definition statement) NCP is to use to select an alternate line to communicate with a device. NCP uses this dial set when it cannot reach that device using the regular nonswitched point-to-point or BSC multipoint line. (This switched network backup function applies only to line operation in network control mode.) Upon request from the host processor, NCP calls the device over a line from the dial set specified.

For NCP to use the alternate line successfully, the lines in the dial set must have the same characteristics as the nonswitched line.

Code a dial set name in DIALALT only if:

- The line represented by this LINE definition statement is a nonswitched point-to-point or BSC multipoint line.

- You want NCP to reach the device over an alternate line when communication with the device over the regular nonswitched line is impossible.

- You specify switched network backup among the options included in NCP (see the SYSCNTRL definition statement).

DIALALT is not valid for an SDLC link (LNCTL=SDLC is specified on the GROUP definition statement).

DIALSET

```
                  ┌─DIALSET=NONE─┐
►►────────────────┴──────────────┴────►◄
       └─DIALSET=─┬─name─┬─┘
                  └─NONE─┘
```
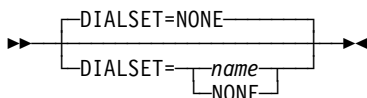
Use: Switched, point-to-point BSC or start-stop lines
Default: NONE

Specifies the name of the dial set (DIALSET definition statement) for this line. Code DIALSET=NONE or omit DIALSET if this line is used only for incoming calls (CALL=IN).

LINES on the DIALSET definition statement must contain the name of this LINE definition statement.

DISABLE
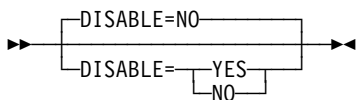
```
                  ┌─DISABLE=NO─┐
►►────────────────┴────────────┴────►◄
       └─DISABLE=─┬─YES─┬─┘
                  └─NO──┘
```

Use: Start-stop and BSC lines in emulation mode
Default: NO

Specifies whether the modem for this line requires a long disable interval when disconnecting from the line. DISABLE=YES provides a 25.6-second interval.

Most modems do not require the long interval, so DISABLE=NO is recommended.

DSCRDDLY
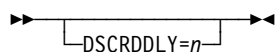
```
►►──────────────────►◄
      └─DSCRDDLY=n─┘
```

Use: 3746 Model 900 frame-relay/ISDN physical line congestion control

Default: Set by the Model 900 based on line speed
Minimum: 0.1
Maximum: 25.4

The value of **n** specifies the DSCRDDLY threshold in seconds. Valid values are in intervals of a tenth of a second.

When a frame is removed from the 3746 Model 900 software queue and put on the 3746 Model 900 hardware queue for transmission, the frame is discarded if the frame has been on the software queue in excess of the amount of time specified by DSCRDDLY.

If the frame has been on the software queue in excess of the amount of time specified by CRATEDLY but less than the amount of time specified by DSCRDDLY, the following action will be taken:

• If more than 100% of the station's COMRATE has been used, the frame is discarded.

• If more than 75% but less than or equal to 100% of the station's COMRATE has been used, the frame is transmitted with the FECN indicator set on.

• If less than 75% of the station's COMRATE has been used, the frame is transmitted.

See the *3745 Communication Controller Models A, 3746 Expansion Unit Model 900: Migration and Planning Guide*, GA33-0183, for more information about the Model 900 congestion control processing.

Note: DSCRDDLY must be greater than or equal to CRATEDLY.

DUALCOM
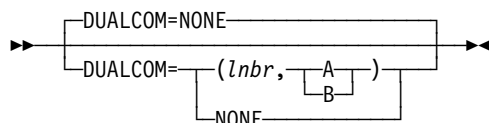
```
             ┌─DUALCOM=NONE────────────────┐
►►───────────┼─────────────────────────────┼──►◄
             └─DUALCOM=──┬─(lnbr,─┬─A─┬─)─┬─┘
                         │        └─B─┘   │
                         └─NONE───────────┘
```

Use: BSC lines in emulation mode
Default: NONE

Specifies whether NCP or EP is to emulate the dual communication interface feature for this line when the line is operating in emulation mode. DUALCOM applies only to a line emulating an IBM 2701 Data Adapter Unit with the dual communication interface feature.

lnbr

> Minimum: 0
> Maximum: 511

> Specifies the *alternate relative line number* of the line formerly attached to the dual communication interface of the IBM 2701.

A|B
> Specifies whether this line (line number specified on ADDRESS on this LINE definition statement) corresponds to dual communication interface A or B.

NONE
> Specifies that the IBM 2701 being emulated is not equipped with the dual communication interface for this line.

DUPLEX

```
                  ┌─DUPLEX=HALF─────┐
►►────────────────┼─────────────────┼──────►◄
                  └─DUPLEX=──┬─HALF─┬┘
                            └─FULL─┘
```
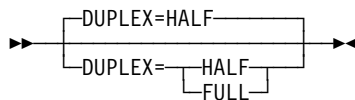
| Use: NCP or EP
| Default: HALF

Specifies whether the communication line and modem constitute a half-duplex or a duplex facility.

Do not confuse this with half-duplex or duplex data transfer. DUPLEX specifies only the physical characteristic of the communication facility, and is used to control the request-to-send (RTS) signal under the following conditions:

BSC and Start-Stop Lines: If you specify DUPLEX=FULL, the RTS signal is active when NCP is either sending or receiving data. If you specify DUPLEX=HALF, the RTS signal is active only when NCP is sending data.

SDLC Lines: If you specify FULL as the second suboperand of ADDRESS, DUPLEX=HALF is not valid. Code DUPLEX=HALF if you specify MODE=SECONDARY on the GROUP definition statement (the default is HALF, and DUPLEX=FULL is not valid). If you specify HALF as the second suboperand of ADDRESS, DUPLEX affects the RTS signal as follows:

- If you specify DUPLEX=FULL, the RTS signal is active when NCP is either sending or receiving data.

- If you specify DUPLEX=HALF, the RTS signal is active only when NCP is sending data.

| **EP Standalone.:** If the DUPLEX keyword is omitted and CUTYPE=2845 or 2848,
| or TERM=2260 or 2265, the default is DUPLEX=FULL; otherwise the
| default is DUPLEX=HALF.

Minimizing Line-Turnaround Delay: Each reversal in the direction of message transmission over an SDLC link causes a line-turnaround delay of several milliseconds if the carrier signal transmitted by the modem is interrupted. Efficiency of the link is therefore lower than if the carrier signal is transmitted continuously, because no message data can be transmitted during these intervals.

Interruption of the carrier signal, and therefore line-turnaround delays, are inherent and unavoidable if the communication facility (including lines and modems) is half-duplex. In a duplex facility, you can prevent line turn-around delays by ensuring that the carrier signal is transmitted continuously. Continuous carrier transmission results from one of the following:

- Continuous activation of the request-to-send signal sent from the communication controller to the modem

- Internally connecting (strapping) the modem to transmit a carrier signal continuously, independent of whether the request-to-send signal is activated.

The request-to-send signal (and therefore the carrier sent by the modem) is activated continuously if:

- The communication facility is duplex.

- The line set to which the SDLC link is attached is duplex.

- You specify FULL as the second parameter of ADDRESS on the corresponding LINE definition statement.

- You code DUPLEX=FULL on the LINE definition statement.

The request-to-send signal is not activated continuously if you specify DUPLEX=HALF on the LINE definition statement. The line-turnaround delay still can be minimized or eliminated, however, if the modem is strapped for continuous carrier operation. Consult the supplier or installer of the modem to determine whether it is capable of continuous carrier operation and, if so, have the modem connected this way.

DYNADMP
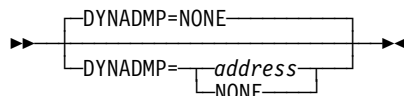
```
                  ┌─DYNADMP=NONE────────┐
►►──────────────┼──────────────────────┼──►◄
                  └─DYNADMP=─┬─address─┬─┘
                            └─NONE────┘
```

Use: Channel links in emulation mode
Default: NONE

Specifies whether the control program (NCP or EP) is to include the dynamic dump facility. DYNADMP also specifies the address of the emulation subchannel over which NCP or EP is to transfer dynamic dump data to the host.

The native subchannel cannot transfer dynamic dump data.

DYNADMP is not valid when you code CA=TYPE7 or CA=TYPE7-TPS or when you are defining ESCON links.

DYNWIND
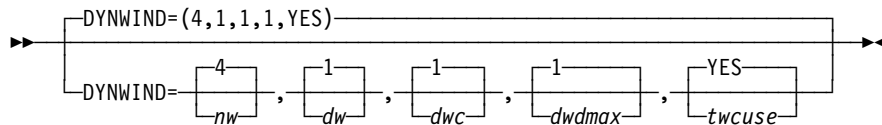
```
          ┌─DYNWIND=(4,1,1,1,YES)──────────────────────────────────┐
►►────────┤                                                        ├────►◄
          │         ┌─4─┐   ┌─1─┐   ┌─1──┐   ┌─1──────┐   ┌─YES───┐ │
          └─DYNWIND=┤   ├,──┤   ├,──┤    ├,──┤        ├,──┤       ├─┘
                    └─nw┘   └─dw┘   └─dwc┘   └─dwdmax─┘   └─twcuse┘
```

Use: Token-Ring, frame-relay, and ISDN physical lines for associated
            logical peripheral stations
            ISDN physical lines for associated logical subarea stations
            Token-Ring and frame-relay logical subarea lines

Default: See text

Specifies the values to be used by NCP to modify the IEEE 802.2 LLC working
window parameter. The values specified increment the size of the working window,
or control how quickly the working window should be closed or dropped when
frame loss occurs. When you code DYNWIND on a physical line definition, the
values specified apply to all logical lines associated with that physical line.

Note: DYNWIND *must* be coded on the PU definition statement for dynamic
reconfiguration and switched PU support via VTAM.

DYNWIND is not valid for NTRI frame handler logical lines.

nw

> Minimum: 1
> Maximum: 8
> Default: 4
>
> Specifies the number of frames that must be received and acknowledged by
> the adjacent device before the sending NCP increments the working window
> value by 1. When the sending NCP receives acknowledgments for *nw* consec-
> utive I-frames, the working window value is incremented, until it reaches the
> value specified by MAXOUT for leased connections, or the value specified in
> the VTAM switched major node for switched connections.

dw

> Default: 1
>
> Specifies the value by which the sending NCP divides or decreases the
> working window value when frame loss occurs.
>
> 1 is a special value which indicates that the working window value should be
> set to 1 when frame loss is detected. Valid **divisor values** are 2, 4, 8, or 16.
> Starting in NCP V7R6, valid **percentage values** are 6, 12, 25, 50, 75, 87, or
> 93. Table 2-62 on page 2-223 summarizes the valid *dw* values.
>
> *dw* is valid only for frame-relay and ISDN lines.

dwc

> Default: 1
>
> Specifies the value by which the sending NCP divides or decreases the
> working window value when backward explicit congestion notification (BECN)
> occurs.

1 is a special value which indicates that the working window value should be set to 1 when BECN is detected. Valid **divisor values** are 2, 4, 8, or 16. Starting in NCP V7R6, valid **percentage values** are 6, 12, 25, 50, 75, 87, or 93. Table 2-62 on page 2-223 summarizes the valid *dwc* values.

dwc is valid only for frame-relay and ISDN lines. See the *NCP and EP Reference* for more information about BECN.

Table 2-62. Working Window Reductions for dw and dwc Values

	Value	Percent Reduction
Special Value	1	N/A
Divisor Values	2	50
	4	75
	8	87
	16	93
Percentage Values	6	6
	12	12
	25	25
	50	50
	75	75
	87	87
	93	93

Note: The default values for *dw* and *dwc* cause the dynamic window algorithm to work according to IEEE 802.2 architecture specifications. Using other valid values allows you to increase bandwidth utilization.

dwdmax

Use: 3745 frame-relay lines
Default: 1 second
Minimum: 0.1 seconds
Maximum: 6 seconds

Specifies the upper limit for the time delay that NCP initiates before transmitting the next I-frame for a FRTE station experiencing congestion. NCP initiates this time delay if congestion continues while a station's working transmit window is equal to 1. If you do not want NCP to initiate the delay, specify a value of 0.

twcuse

Use: 3745 frame-relay lines
Default: YES

Specifies whether or not an NCP FRTE station ignores any subsequent BECN for the window congestion time of 100 milliseconds (ms) after receiving the initial BECN. Specify YES to have a station ignore subsequent BECNs for 100 ms if the working transmit window is adjusted or NO to have the station react to all BECNs. This suboperand is intended to keep NCP stations from overreacting to mild congestion.

For a more complete description of coding DYNWIND, see the *NCP, SSP, and EP Resource Definition Guide*.

ETRATIO

```
                 ┌─ETRATIO=30──┐
►►───────┼──────────────┼───────►◄
                 └─ETRATIO=ratio─┘
```
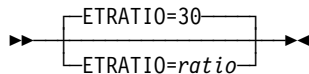
Default: 30
Minimum: 1
Maximum: 255

Specifies the error-to-transmission ratio (the ratio of the number of errors to the number of transmissions). This ratio is used by programs that analyze statistical maintenance records (the NetView program does not use ETRATIO).

ETRATIO specifies this ratio in 10ths of a percent; for example, ETRATIO=125 indicates a ratio of 12.5%.

ETRATIO is not valid when you code LEVEL5=USER on the GROUP definition statement or when you are defining ESCON links.

FEATURE

```
►►──────────────────────────────────────────────►◄
     └─FEATURE=(─┤ Terminal Features ├─(1)─)─┘
```

Terminal Features:

```
   ┌─NODUALCD─┐   ┌─NOIMEND─┐   ┌─LRC──┐   ┌─NOSPACE─┐   ┌─NSTXBCC─┐
├──┼──────────┼─,─┼─────────┼─,─┼──────┼─,─┼─────────┼─,─┼─────────┼──┤
   ├─DUALCODE─┤   ├─IMEND───┤   ├─LRC──┤   ├─SPACE───┤   ├─STXBCC──┤
   └─NODUALCD─┘   └─NOIMEND─┘   └─NOLRC┘   └─NOSPACE─┘   └─NSTXBCC─┘
```
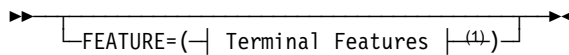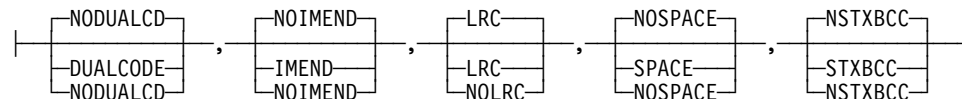
1 Order of Terminal Features does not matter.

Use: BSC or start-stop terminal lines (NCP or EP)
Example: FEATURE=(DUALCODE,NOLRC,SPACE)

Specifies which BSC and start-stop terminal features NCP or EP is to support. You can code the suboperands in any order. Separate the suboperands by commas.

Note: This is the lowest-level definition statement on which you can code FEATURE for emulation mode. For network control mode, you can code FEATURE on the CLUSTER and TERMINAL definition statements.

NODUALCD│DUALCODE

Default: NODUALCD

Specifies whether the program emulates the dual code feature of an IBM 2701. The dual code feature allows message transmission over the line in either EBCDIC or ASCII. Specify DUALCODE if either code can be used and the access method can change the code. Specify NODUALCD if the dual code function is not required.

NOIMEND│IMEND

Default: NOIMEND

Specifies whether a receive operation from a start-stop terminal immediately ends when the EOT is received. If you code IMEND, the operation ends imme-

diately. If you code NOIMEND, the program delays ending the receive opera-
tion until the line becomes electrically quiet following receipt of the EOT. The
absence of further characters indicates that the EOT is valid. When you
specify IMEND, LNQTCNT (LINE definition statement for network control mode)
or QUIETCT (GROUP definition statement for emulation mode) is ignored.

LRC│NOLRC

Default: LRC

Specifies whether the start-stop terminals on this line are equipped with recor-
d-checking capability, either as an inherent function or as a feature. Specify
LRC for record-checking capability or NOLRC for no record-checking capability.
Table 2-63 lists the valid stations for LRC.

Table 2-63. Device Types Capable of Record Checking

Station	TERM keyword specifies
IBM 1050	1050
IBM 1060	1060
IBM 2260	2260
IBM 2265	2265
IBM 2740 Model 1	2740-1
IBM 2740 Model 2	2740-2
IBM System/7	SYS7

NOSPACE│SPACE

Default: NOSPACE

Specifies whether the control program reacts to space characters received from
WTTY terminals as downshift characters when the line is in emulation mode. If
you specify SPACE, each space character received from a terminal causes the
program to send all subsequent text characters to the host processor in their
downshifted form.

If you specify NOSPACE (or omit the suboperand), the program does not
convert the characters to their downshifted form but sends them as received
from the terminal.

NSTXBCC│STXBCC

Default: NSTXBCC

Specifies whether SWIFT support is included in the control program. If the con-
troller supports devices on this line that are equipped with the SWIFT feature,
specify FEATURE=STXBCC. If the devices on this line are not equipped with
the SWIFT feature, specify FEATURE=NSTXBCC or omit the suboperand.

FRAMECNT

```
                  ┌─FRAMECNT=(100000,5000)──────────┐
►►────────────────┴─────────────────────────────────┴─────►◄
       │                  ┌─100000─┐       ┌─5000─┐      │
       └─FRAMECNT=(───────┼────────┼───,───┼──────┼──)───┘
                          └───m────┘       └──n───┘
```
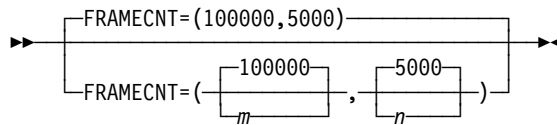
Use: Ethernet-type LANs
Examples: FRAMECNT=(65535,4095)
             FRAMECNT=(,2000)

Specifies threshold values for the total number of frames transmitted and received
and for the total number of frames that can be lost because of transmission and
receive errors during an error cycle. If either threshold is reached, NCP generates
an alert. See the *NCP and EP Reference* for more information on NCP statistical
counters.

m

> Minimum: 1
> Maximum: 9999999
> Default: 100000

> *m* is the threshold for both the total number of frames transmitted and the total
> number of frames received.

n

> Minimum: 1
> Maximum: 99999
> Default: 5000

> *n* is the threshold for both the total number of frames lost due to transmission
> errors and the total number of frames lost due to receive errors.

Attention: Specifying small values for FRAMECNT can degrade NCP performance
and can cause NCP to generate an excessive number of alerts. For normal opera-
tion, values greater than 300 are recommended.

FRRATES

```
                  ┌─FRRATES=(1,1,1)──────────────────────────────────────────┐
──►►──────────────┤                                                          ├──►◄──
                  │            ┌─1─────────┐   ┌─1────────┐   ┌─1────────┐    │
                  └─FRRATES=(──┤           ├─,─┤          ├─,─┤          ├─)──┘
                               └─sna_priority─┘   └─ip_priority─┘   └─fh_priority─┘
```
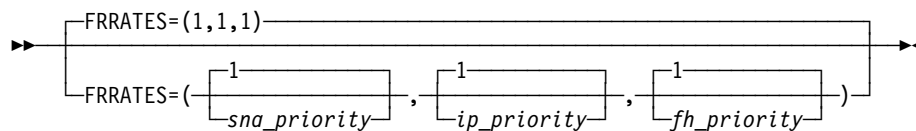
Use: 3745 frame-relay physical lines
Default: (1,1,1)
Minimum: 1
Maximum: 64

Specifies the relative transmission priority for the SNA, IP, and frame handler pro-
tocol traffic when supported by the physical line. The value of each protocol's *pri-
ority* suboperand is multiplied by the value of the DATABLK keyword on the
physical LINE definition statement (*priority*∗DATABLK) to determine the guaranteed
minimum number of bytes provided for a protocol for the transmission of associated
traffic before a different protocol is provided a transmission opportunity.

FRSPRI

```
        ┌─FRSPRI=0─┐
──►►────┤          ├──►◄──
        └─FRSPRI=n─┘
```

Use: 3746 Model 900 congestion control
Default: 0
Minimum: 0
Maximum: 3

FRSPRI specifies a relative service priority at a station level. The lowest priority is
0, and the highest priority is 3.

When congestion occurs on a 3746 Model 900 attached Frame Relay/ISDN phys-
ical link, the 3746 Model 900 will use the relative service priority to determine the
order in which stations will be serviced during the next transmit period.

The following section lists the stations that the FRSPRI parameter supports:

- Frame Relay Frame Handler PUs
- Frame Relay Logical Subarea PUs
- Frame Relay Logical Peripheral PUs
- ISDN Logical Subarea PUs (LOGical,SUB)
- ISDN Logical Peripheral PUs (LOGical,PER)

Note: A frame handler added by a dynamic reconfiguration (DR) definition inherits
its FRSPRI value from the FRSPRI value of the physical LINE to which the frame
handler PU is added.

See the *3745 Communication Controller Models A, 3746 Expansion Unit Model
900: Migration and Planning Guide*, GA33-0183, for more information about the
Model 900 congestion control processing.

HDXSP

```
                     ┌─HDXSP=NO──────┐
►►───────────────────┼───────────────┼───►◄
                     └─HDXSP=──┬─YES─┘
                              └─NO──┘
```
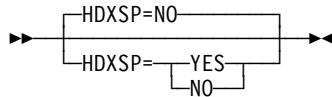
Use: SDLC lines
Default: NO

Specifies whether sending data to a terminal has priority over polling. HDXSP
applies only to SDLC links that have one line interface address (half-duplex) and
have more than one active device.

HDXSP=YES lets NCP send, at the first opportunity, outgoing data that is ready for
transmission without having to wait until it is the destination terminal's time to be
polled.

HDXSP=NO lets NCP send the outgoing data when it is the destination terminal's
turn to be polled. In this case, NCP sends the data just before sending the polling
sequence unless the link is in the poll-wait state. See PAUSE on the LINE defi-
nition statement on page 2-260 for a description of poll wait.

HDXSP is valid only if you specified LNCTL=SDLC on the GROUP definition state-
ment, PAUSE=0 on LINE, and a single line interface address in ADDRESS on the
LINE definition statement.

Casual Connection: Code HDXSP=NO if you specify MODE=SECONDARY on
the GROUP definition statement.

HICHAN

```
                     ┌─HICHAN=NONE──────┐
►►───────────────────┼──────────────────┼───►◄
                     └─HICHAN=──┬─address─┐
                              └─NONE────┘
```
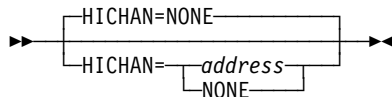
Use: Channel links in emulation mode
Default: NONE

Specifies the highest subchannel address on the channel adapter associated with
any line operating in emulation mode, or the address of the subchannel used for
the dynamic dump facility. *address* must equal or exceed the highest emulation
subchannel address specified for ADDRESS on the LINE definition statement.
HICHAN is required if you code NCPCA=INACTIVE or if you include emulation
functions.

HICHAN is not valid when you code CA=TYPE7 or CA=TYPE7-TPS or when you
are defining ESCON links.

The value of *address* can range from X'00' to X'FF' but must be equal to or
greater than the value coded on LOCHAN.

HISPEED

```
►►──────────────────────────►◄
    └─HISPEED=─┬─YES─┬─►◄
              └─NO──┘
```
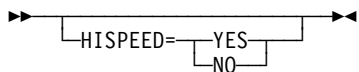
Use:	BSC and TSS SDLC and frame-relay lines in network control mode
Default:	See text

Specifies whether this line operates at speeds of 144 Kbps or greater. HISPEED is valid only for TSS lines (line addresses 0–895) for which TYPE=NCP and LNCTL=SDLC or LNCTL=BSC.

When the value of SPEED on this LINE definition statement is 144000 or greater, the default is HISPEED=YES; otherwise, the default is HISPEED=NO.

If you code HISPEED=YES, the ADDRESS keyword on this definition statement may specify any address on the scanner, but only one line in the scanner can operate at once.

You must code HISPEED=YES to operate a 144-Kbps or greater line attached to a TSS adapter. If you do not, the line remains pending contact and no error message is issued.

HLOQSTOP
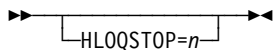
```
►►───────────────►◄
    └─HLOQSTOP=n─┘
```

Use: 3746 Model 900 frame-relay congestion control

Default:	Set by the Model 900 based on line speed
Minimum:	1
Maximum:	254

The value of **n** is multiplied by the value of the DATABLK keyword to determine the threshold in bytes.

Specifies the threshold that determines the maximum number of bytes that can be queued to a 3746 Model 900 attached frame relay physical line's hardware queue before a 3746 Model 900 software queue is created for the physical line. See the *3745 Communication Controller Models A, 3746 Expansion Unit Model 900: Migration and Planning Guide*, GA33-0183, for more information about the Model 900 congestion control processing.

HOSTLINK

```
►►───────────────►◄
    └─HOSTLINK=n─┘
```
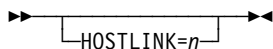
Use:	Required for ESCON logical links
Default:	None
Minimum:	1
Maximum:	16

Assigns an identifier to this logical line so that it can be associated with a host link using the MOSS-E ESCA Link Information panel.

ID

```
►►──┬──────────────┬──►◄
    └─ID=nodeid────┘
```
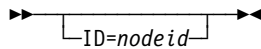
Use:	Switched SDLC lines, ISDN B-channel lines, and switched frame-relay physical lines.
Default:	None

Specifies, in hexadecimal representation, the node identification (NODEID) field to be returned to the host in the request contact request/response unit when a dial connection is made. Eight hexadecimal numbers must be specified. The resulting number must be unique within the network and must correspond with the IDNUM and IDBLK keywords of the PU definition statement specified in the VTAM switched major node definition.

Attention: NCP's request contact processing will substitute your specified NODEID for the NODEID in the received XID before the XID is passed to the host for station identification. If you specify this keyword, network awareness terminates at the port. The security previously provided by host verification of the station's NODEID is not possible. Therefore, the burden of security resides at the application level.

All stations that call in on this line are identified by the same NODEID. You should make sure that the specified NODEID is unique within your network.

ID is not valid if you code DIAL=YES and PUTYPE=4 for switched subarea support lines.

With the exception of ISDN B-channel lines, ID is not valid if you code AUTOCOPY on the LINE definition statement.

Switched Frame-Relay Physical Lines: ID is required if DIAL=YES on the GROUP statement.

INBFRS

```
       ┌─INBFRS=3──────┐
►►──┼───────────────┼──►◄
       └─INBFRS=buffers─┘
```
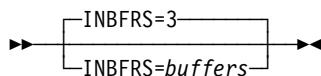
Use:	Channel links
Default:	3
Minimum:	3
Maximum:	128

| Specifies how many controller buffers NCP or EP is to allocate for the first data
| transfer received from the host processor. For subsequent data transfers, NCP or
| EP adjusts the number of buffers dynamically.

This keyword is ignored when you code CA=TYPE7 or CA=TYPE7-TPS.

ESCON Links: The INBFRS keyword is not valid.

| **EP Standalone:** The INBFRS keyword is valid only when NCPCA=ACTIVE is
| specified.

INTFACE

```
►►──┬─────────────────────────────┬──►◄
     └─INTFACE=(name─┬───────┬─)─┘
                      └─,mtu─┘
```

Use: Internet Protocol

Specifies an internet router interface name, which must be unique within your NCP generation. The interface name defined is referenced on the IPLOCAL and IPROUTE definition statements to correlate internet route table entries with an internet connection.

Frame Relay, Channel, and Internal NCP to Model 900 Point-to-Point IP Connections: INTFACE is coded on the PU definition statement for channel IP connections, frame-relay IP connections, and internal NCP to Model 900 point-to-point IP connections. Refer to INTFACE on the PU definition statement on page 2-383 for information on coding INTFACE.

name

   Default: None

   Specifies the interface name, which must be unique and no more than 8 characters long.

mtu

   Default: See text
   Minimum: See text
   Maximum: See text

   Specifies the maximum transfer unit (MTU), which is the maximum size, in bytes, of the transmission frame.

   Ethernet-Type LAN: The default, minimum, and maximum values for *mtu* are determined by the value of LANTYPE on the PU definition statement. Table 2-64 shows the default, minimum, and maximum values for *mtu*.

   Table 2-64. Values for the mtu Suboperand of INTFACE on the LINE Definition Statement

   | When You Code | The Value Is | | |
LANTYPE=	ENETV2	802.3	DYNAMIC
Default	1500	1492	1492
Minimum	68	68	68
Maximum	1500	1492	1492

   NTRI: When you code ECLTYPE=PHYSICAL on the LINE definition statement and NETWORK=IP on a PU definition statement following this LINE definition statement, the default value for *mtu* is 2048 or the value coded for MAXTSL minus 4, whichever is less. The minimum value is 68 and the maximum value is MAXTSL minus 4.

   Ensure that the value you specify for *mtu* is less than the largest frame size supported by any Token-Ring source routing bridges in the routes

between NCP and remote internet hosts. Otherwise, NCP will not
accept ARP requests or replies from the remote hosts and consequently
will not be able to route data to them.

IPL

```
           ┌─IPL=NO──────────┐
►►─────────┼─────────────────┼────►◄
           └─IPL=─┬─YES─┬─────┘
                  └─NO──┘
```
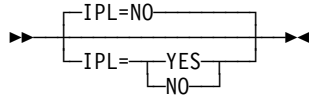
Default: NO

Specifies whether this NCP can be loaded and dumped over this SDLC link. Any
SDLC link defined with IPL=YES must agree with the IPL configuration data set.
IPL is valid only for links that attach to type 4 physical units. Only a maximum of
eight LINE definition statements can specify IPL=YES in a single NCP generation
(not including Token-Ring, frame-relay, ISDN, 3746 Model 900 SDLC, and switched
X.21 lines). If you code IPL=YES, commands from the access method are
accepted over this link that cause NCP to abnormally end and then be reloaded
with another NCP. IPL is not valid if you code VIRTUAL=YES in the GROUP defi-
nition statement for this line.

If you code IPL=YES on a switched or nonswitched SDLC line or on a nonswitched
X.21 line, you must code MONLINK=YES or MONLINK=CONTINUOUS at the
same relative address for the line of the load module replacing this load module.

IPL=YES is not valid if you specify MODE=PRIMARY on the GROUP definition
statement and you do not code SDLCST on the LINE definition statement for
primary multipoint subarea links.

Switched Subarea Lines: You may code IPL=YES if DIAL=YES and
LNCTL=SDLC on the GROUP definition statement.

Channel Links: The IPL keyword is not valid for channel links (LNCTL=CA on the
GROUP definition statement). 390 OEMI channels operate as though you
specified IPL=YES.

Token Ring: If you specify ECLTYPE=(LOGICAL,SUBAREA), you may specify
IPL=YES.

Frame Relay: If you specify FRELAY=LOGICAL or
FRELAY=(LOGICAL,SUBAREA), you may specify IPL=YES.

ISDN: If you specify ISDN=(LOGICAL,SUBAREA), you can specify IPL=YES.

ESCON Links: The IPL keyword is not valid for ESCON links. To load or dump
NCP over an ESCON link, select the MOSS-E IPL option. Also code
MONLINK=YES or CONTINUOUS on this LINE definition statement and
XMONLNK=YES on the PU definition statement for the link station.

ISDNCALL
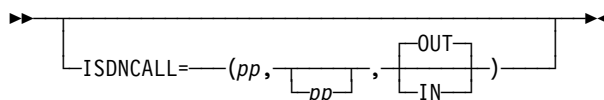
```
►►─────────────────────────────────────────────►◄
   └─ISDNCALL=──(pp,─┬─────┬─,─┬─OUT─┬─)─┘
                     └─pp──┘   └─IN──┘
```

| Use: 3746 Model 900 nonswitched frame-relay physical lines

Specifies information for ISDN backup of the 3746 Model 900 frame-relay physical line.

pp
>   Indicates one or two ISDN physical lines to use for backup of the 3746 Model 900 frame-relay physical line. *pp* specifies the port address defined by the PORTADD keyword on the ISDN physical LINE definition statement. If you specify two ISDN lines, the first one specified is used if it is active and has available resources; otherwise, the second is used if it is active and has available resources.

IN|OUT

>   Default: OUT

>   Indicates which party in the connection initiates the backup. OUT indicates that the local (calling) party will initiate the connection for the frame-relay backup. IN indicates that the local party should wait for the remote party to initiate the connection for the frame-relay backup. To prevent both parties in a connection from trying to initiate the backup, if one specifies OUT the other should specify IN.

ISDNLPN

►►──┬─────────────────┬──►◄
      └─ISDNLPN=*lpn*─┘

| Use: ISDN physical lines and 3746 Model 900 nonswitched frame-relay
| physical lines

Specifies the local calling party number for the ISDN line that you are defining or indicating for 3746 Model 900 frame-relay physical line backup. The value you specify is the ISDN address for the physical interface.

lpn must be defined according to the ITU-T Q.931 recommendation for a calling party number. The format of *lpn* is:

tnps3d3d3d3d3d3d3d3d3d3d3d3d3d3d3d3d

or

tn3d3d3d3d3d3d3d3d3d3d3d3d3d3d3d3d

where:

t   is the type of number (0, 1, 2, 3, 4, 8, 9, A, B, or C) where:

>   0 or 8   indicates an unknown number

>   1 or 9   indicates an international number

>   2 or A   indicates a national number

>   3 or B   indicates a network specific number

>   4 or C   indicates a subscriber number

>   *t* must be greater than 7 when coded on a 3746 Model 900 frame-relay physical line.

> *n*   is the numbering plan identification (0, 1, 8, or 9) where:
>
> > 0   indicates an unknown numbering plan
> >
> > 1   indicates the ISDN telephone numbering plan (E.164)
> >
> > 8   indicates the national standard numbering plan
> >
> > 9   indicates a private numbering plan
>
> *p*   is the presentation indicator (8 or A) where:
>
> > 8   specifies that the network include the calling party number and subaddress in the setup message sent to the called DTE.
> >
> > A   specifies that the network not include the calling party number and subaddress in the setup message sent to the called DTE.
> >
> > Specifying this value is optional. If *t* is less than or equal to 7, *p* is present; if *t* is greater than 7, *p* is not present. If p is not present, the network will include the calling party number and subaddress in the setup message sent to the called DTE.
>
> *s*   is the screening indicator (0). Specifying this value is optional. If *t* is less than or equal to 7, *s* is present; if *t* is greater than 7, *s* is not present. If specified, this value must be 0.
>
> *d*   is a decimal digit of the calling party number in the range 0–9. Each *d* is preceded by a 3. There can be from one to fifteen digits specified (each preceded by a 3).
>
> An example is ISDNLPN=0180303132.

ISDNLSA

```
►►──┬──────────────┬──►◄
    └─ISDNLSA=lsa──┘
```

| Use: ISDN physical lines and 3746 Model 900 nonswitched frame-relay physical lines

Specifies the local calling party subaddress for the ISDN line that you are defining or indicating for frame-relay line backup. The value you specify is the ISDN subaddress for the physical interface.

lsa must be defined according to the ITU-T Q.931 recommendation for a calling party subaddress. The format of *lsa* is:

tpdddd

where:

> *t*   is the type of subaddress (8 or A) where:
>
> > 8   indicates a network service access point (NSAP) address, which must be coded using preferred binary encoding as described in the ISO 8348 standard, Addendum 2 or ITU-T recommendation X.213.
> >
> > A   indicates a user-specified subaddress
>
> *p*   is the odd or even indicator (0 or 8). It is significant only when *t* is A.

d   is a digit of the subaddress. The subaddress should be 2–40 digits in length with an even number of digits. When *t* is 8, the last digit is not significant and must be 0.

An example is A0345678.

ISDNRPN

```
►►─────────────────►◄
    └─ISDNRPN=rpn─┘
```

| Use: 3746 Model 900 nonswitched frame-relay physical lines

Specifies the remote called party number of the backup ISDN line for the 3746 Model 900 frame-relay physical line. ISDNRPN is required when ISDNCALL=OUT is specified or taken as the default.

rpn must be defined according to the ITU-T Q.931 recommendation for a called party number. The format of *rpn* is

tn3d3d3d3d3d3d3d3d3d3d3d3d3d3d3d

where:

t   is the type of number (8, 9, A, B, or C) where:

         8   indicates an unknown number

         9   indicates an international number

         A   indicates a national number

         B   indicates a network specific number

         C   indicates a subscriber number

n   is the numbering plan identification (0, 1, 8, or 9) where:

         0   indicates an unknown numbering plan

         1   indicates the ISDN telephone numbering plan (E.164)

         8   indicates the national standard numbering plan

         9   indicates a private numbering plan

d   is a decimal digit of the called party number in the range 0–9. Each *d* is preceded by a 3. There can be from one to fifteen digits specified (each preceded by a 3).

An example is ISDNRPN=80303132.

ISDNRSA

```
►►─────────────────►◄
    └─ISDNRSA=rsa─┘
```

| Use: 3746 Model 900 nonswitched frame-relay physical lines

Specifies the remote called party subaddress of the backup ISDN line for the 3746 Model 900 frame-relay physical line.

rsa must be defined according to the ITU-T Q.931 recommendation for a called party subaddress. The format of *rsa* is:

tpdddd

where:

t   is the type of subaddress (8 or A) where:

        8   indicates a network service access point (NSAP) address, which must be coded using preferred binary encoding as described in the ISO 8348 standard, Addendum 2, or ITU-T recommendation X.213.

        A   indicates a user-specified subaddress

p   is the odd or even indicator (0 or 8). It is significant only when *t* is A.

d   is a decimal digit of the subaddress. The subaddress should be 2–40 digits in length with an even number of digits. When *t* is 8, the last digit is not significant and must be 0.

LIMRES

```
          ┌─LIMRES=NO──────┐
►►────────┼────────────────┼──►◄
          └─LIMRES=──┬─YES─┘
                     └─NO──┘
```
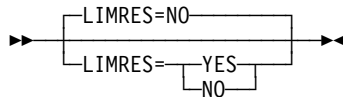
Default: NO

Specifies whether or not a line is for a limited resource session that should be deactivated automatically when session traffic ceases on a switched link. LIMRES is valid only for peripheral links.

LINECB
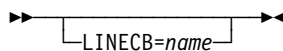
```
►►──┬──────────────┬──►◄
    └─LINECB=name──┘
```

Use: User-written code and Internet Protocol
Default: None

Specifies the name of a user-defined control block associated with a virtual link. LINECB is valid only if you code VIRTUAL=YES on the GROUP definition statement.

Internet Protocol: When NCST=IP is specified on the GROUP definition statement, only LINECB=CXSNLK is valid. If you do not code LINECB, NDF adds LINECB=CXSNLK to the generation definition.

LINEFVT

```
►►──┬───────────────┬──►◄
    └─LINEFVT=name──┘
```
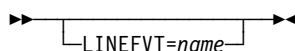
Use: User-written code and Internet Protocol
Default: None

Specifies the name of the function vector table associated with a virtual link. LINEFVT is required if you specify VIRTUAL=YES on the GROUP definition statement. This keyword is not valid if VIRTUAL=NO. An EXTRN statement for each unique FVT symbol must be included in the user source code that is copied into the tables during assembly. (See SRCHI on the GENEND definition statement on page 2-93.)

The function vector table must be preassembled and included in the user object modules. (See INCHI on the GENEND definition statement on page 2-96.)

Internet Protocol: When NCST=IP is specified on the GROUP definition statement, only LINEFVT=CXSXFVT is valid. If you do not code LINEFVT, NDF adds LINEFVT=CXSXFVT to the generation definition.

LINESIZ
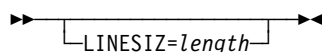
```
►►─┬──────────────────┬─►◄
   └─LINESIZ=length─┘
```

Use: Start-stop lines in network control mode
Default: See text
Minimum: 1
Maximum: 255

Specifies the length of the print line for printer-type devices on this line.

If you omit LINESIZ, a line length of 72 print positions is the default for TWX and WTTY terminals. The default is 130 print positions for all other types of terminals. Omit LINESIZ if the line is a multiple-terminal-access line.

LNQTCNT

```
      ┌─LNQTCNT=1──────┐
►►─┼────────────────┼─►◄
      └─LNQTCNT=count─┘
```
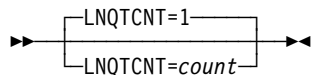
Use: Start-stop lines in network control mode
Default: 1

Specifies the number of times NCP tests a start-stop line after receiving data to ensure that the line has become electrically quiet. The line must become electrically quiet before the next data transmission begins or loss of message data can occur.

The default value of 1 is appropriate for most start-stop lines under 1200 bps. For lines operating at 1200 bps or more, several tests in succession may be necessary to ensure that the line is quiet. The recommended value for 1200-bps start-stop lines is 5.

LOCADD
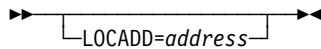
```
►►──┬──────────────────┬──►◄
     └─LOCADD=address──┘
```

Use: Token-Ring, frame-relay BAN connections, ISDN peripheral BAN connections, and Ethernet-type LAN physical lines

Default: None

Specifies the site-assigned address for a line connected to a LAN.

Token Ring: LOCADD is required when ECLTYPE=PHYSICAL is specified on the GROUP definition statement. LOCADD specifies a 6-byte locally administered address for the TIC.

A locally administered MAC address coded for LOCADD cannot have the same value as that specified on ADDR on the PU definition statement.

For duplicate TIC addressing, you may code identical LOCADD values for a set of physical lines:

- For NCP V7R4 and earlier releases, coding identical LOCADD values is valid only when you code ECLTYPE=(PHYSICAL,PERIPHERAL) on the GROUP definition statement and neither physical line supports NTRI frame handler logical links.

- Starting in NCP V7R5, coding identical LOCADD values is valid if you code ECLTYPE=(PHYSICAL,PERIPHERAL|SUBAREA|ANY) on the GROUP definition statement. However, only one of the physical lines may support NTRI frame handler logical links.

If you code different or unique LOCADD values, the associated PORTADD values must be different.

The valid format for LOCADD is *xyyyyyyyyyyy*, with the following restrictions:

- *x* must be a number in the range 4–7.
- Each *y* must be a number in the range 0–X'F'.
- LOCADD=400000000000 is not valid.

3746 Model 900 Token Ring: If you define a 3746 Model 900 Token-Ring line at relative line number 2080 (ADDRESS=2080 on the LINE definition statement), the value of LOCADD must match the MAC address of the TIC. You can obtain this address from MOSS-E.

Ethernet-Type LANs: LOCADD specifies the site-assigned address of the Ethernet-type LAN interface. This address must be unique among local and remote LANs to which this adapter interface is attached. If this keyword is not specified, the hardware address of the ESS adapter interface is used for this keyword value.

The format you use is LOCADD=*xyyyyyyyyyyy*. The value of *x* must be a number from 4 to 7; the value of each *y* must be a number from 0 to X'F'. Neither LOCADD=400000000000 nor LOCADD=7FFFFFFFFFFF is valid.

Frame Relay: LOCADD specifies the NCP virtual MAC address used by remote BAN routers for frame-relay connections. From a BAN router's standpoint, the virtual MAC address is the boundary node identifier. The value speci-

fied for LOCADD must be the same in all routers that will have BAN connections to NCP for all DLCIs over this physical line.

The default for LOCADD is X'4FFF00000000'. The range is X'000000000000' to X'7FFFFFFFFFFF'.

ISDN: LOCADD specifies the NCP virtual MAC address used by remote BAN routers for ISDN peripheral connections. From a BAN router's standpoint, the virtual MAC address is the boundary node identifier. The value specified for LOCADD must be the same in all routers that will have BAN connections to NCP for all DLCIs over this physical line.

The default for LOCADD is X'4FFF00000000'. The range is X'000000000000' to X'7FFFFFFFFFFF'.

LOCALTO
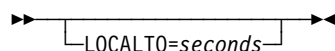
```
►►──┬──────────────────┬──►◄
    └─LOCALTO=seconds──┘
```

Use: Token-Ring logical subarea lines
           Frame-relay logical subarea lines
           Frame-relay logical peripheral lines
Default: See text
Minimum: 0.6
Maximum: 20.0

Specifies the reply timer (T1) value in seconds. You can also specify this value using T1TIMER coded on the following definition statements:

- BUILD for Token-Ring logical,peripheral resources and NTRI frame handler logical lines
- PU for Token-Ring logical subarea resources
- PU for frame-relay logical resources

Token Ring: LOCALTO specifies the Token-Ring reply timer (T1) value for stations connected to a local Token Ring. LOCALTO is valid only for logical subarea lines. (Code LOCALTO on the BUILD definition statement to specify a T1 value for logical peripheral lines and frame handler logical lines.)

The value of LOCALTO must not be greater than the value of REMOTTO on this LINE definition statement. The default value for LOCALTO is either the value of REMOTTO on this LINE definition statement or the value of LOCALTO on the BUILD definition statement, whichever is less.

Frame Relay: LOCALTO specifies the frame-relay reply timer value for stations connected to a frame-relay network. If you do not code LOCALTO on either the LINE or GROUP definition statement, the default is 6.0. The value of LOCALTO on the BUILD definition statement is not used for frame relay.

Important: It is important to code a value for LOCALTO that reflects the maximum amount of time that it takes to receive a solicited response from the adjacent link station. Factors that can affect this time include line speed, propagation delay, maximum frame size, delay caused by intermediate frame handlers, and the number of DLCIs supported by the associated physical line that could create queuing delays for the FRTEs. If the value

you code for LOCALTO is too small, the FRTEs can lose synchronization and fail the connection. This is indicated by an INOP 01 and a link event or an alert that indicates a frame reject (FRMR) condition. If this occurs, code a larger LOCALTO value for this FRTE connection.

If a substitute route for a PVC is defined in the frame-relay network, you may need to code the values of LOCALTO and RETRIES such that the combination of these two values is greater than the LMI error threshold of any node in the PVC. This gives the network time to switch to the substitute PVC.

An indication of INOP 02 can occur for a frame-relay logical line if it is used to perform a remote load. To avoid the condition where the INOP 02 occurs prior to the +RSP(IPLFINAL), the LMI error threshold time interval for the link station subport should be at least 3 times the T1 timer interval for the logical subarea line. The LMI error threshold time interval is $t391 \times n392$, where $t391$ is a time interval defined by TIMERS on the PU definition statement for the link station subport, and $n392$ is an error count threshold defined by ERRORT on the same PU definition statement.

ISDN: The LOCALTO keyword is not valid for ISDN resources. Use the T1TIMER keyword.

LOCHAN
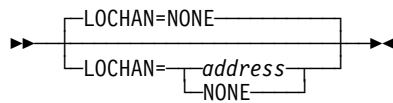
Use: Required for channel links in emulation mode
Default: NONE

Specifies the lowest subchannel address on the channel adapter associated with any line operating in emulation mode. LOCHAN is required if NCPCA=INACTIVE is coded or the program includes emulation functions. LOCHAN is not valid when you code CA=TYPE7 or CA=TYPE7-TPS. The value of *address* can range from X'00' to X'FF'.

ESCON Links: The LOCHAN keyword is not valid.

LPDATS
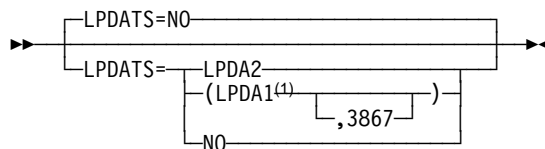
[1] Valid only for TSS- and HPTSS-attached lines

Use: BSC and SDLC lines with externally clocked modems that support the LPDA facility
Default: NO

Specifies whether the line uses modems that support LPDA commands, and whether the modems operate in normal (LPDATS=LPDA2) or migration

(LPDATS=LPDA1 or LPDATS=LPDA1,3867) mode. Code CLOCKNG=EXT on the LINE definition statement to indicate externally clocked modems.

Code LPDATS=NO or omit this keyword under the following conditions:

- If you specify LEVEL5=USER on the GROUP definition statement
- If you are using casual connection and have coded MODE=SECONDARY on the GROUP definition statement
- For Ethernet-type LAN connections
- For frame-relay lines
- For HPTSS lines
- For Token-Ring lines
- For X.21 lines

Code LPDATS=LPDA2 for a link attached to channel A of an IBM 786x modem. LPDA2 capability is allowed on all ports of IBM 786x modems. You must define an LPDA2 port A line if you code LPDA2 on any port other than port A. You may not code LPDA1 and LPDA2 lines with the same CORNUM.

- For IBM 786x modems extended LPDA2 capability connected to port A, code the following:

  ```
  CHNLZ=YES
  CORNUM=x
  LPDATS=LPDA2
  PORT=A
  ```

- For IBM 786x modems extended LPDA2 capability connected to a port other than A, code the following (where *n* is B, C, or D):

  ```
  CHNLZ=YES
  CORNUM=x
  LPDATS=LPDA2
  PORT=n
  CHANLA=NO
  ```

You can code LPDATS=LPDA2 for a line and run LPDA2 tests for the primary side (MODE=PRIMARY) or secondary side (MODE=SECONDARY) of a subarea link. However, multiple errors can result if you initiate an LPDA2 test from both sides of a subarea link at the same time.

When you want to run LPDA2 tests from the secondary side, first issue a Set Link Attributes command to the primary side of the link to stop its tests (if they are currently permitted), then issue a Set Link Attributes command to the secondary side of the link to allow the tests.

To originate LPDA2 tests from either the primary or secondary side of a link, you must configure the modem that is local to that side as primary.

Attention: NCP might initiate unsolicited LPDA2 tests. If both the primary and secondary sides of a line are defined as LPDATS=LPDA2, multiple errors can result. To prevent these errors, code LPDATS=LPDA2 on either the primary or the secondary side of a subarea link, but not on both sides at the same time. NCP does not test for this error condition.

Table 2-65 on page 2-242 shows which modems are valid depending on which suboperand you choose.

Table 2-65. Values for LPDATS on the LINE Definition Statement

If you have this type of modem	And it runs in this mode	Code
3833 3834 3837 3863 3864 3865 3868	Migration	LPDATS=LPDA1
5865 5866 5868	Migration	LPDATS=LPDA1
3867	LPDA1	LPDATS=(LPDA1,3867)
5865 5866 5868	Normal	LPDATS=LPDA2
7861[1] 7868[1]	Migration	LPDATS=LPDA1 CHNLZ=NO
7861 7868	Normal	LPDATS=LPDA2
7825 7865	Normal	LPDATS=LPDA2

[1] The 7861 and 7868 modems cannot operate as channelized modems when in migration mode.

Note: Do not code LPDATS=LPDA1 for a 586x modem if that modem has a two-wire switched network backup (SNBU) coupler installed and the SNBU capability is going to be used.

When you specify that a line uses modems that support LPDA commands, be aware that when an LPDA test is running on that line, the remote devices[11] are not polled. Therefore, make the remote device time outs (the time a device waits to be polled before going off-line) large enough to allow the LPDA tests to be run. Note that with a multipoint configuration it is possible for LPDA tests to be run on two or more successive devices, during which time none of the other remote devices are polled. Therefore, with multipoint configurations, larger device time out values may be required than with point-to-point configurations. LPDATS is not valid for high-speed scanner (HSS) lines on the IBM 3745 Communication Controller.

Point-to-Point Configurations: Device time out values should be greater than 20 seconds for LPDA1 lines. Device time out values should be greater than 30 seconds plus the value of ENABLTO on the BUILD definition statement for LPDA2 lines. This prevents the possibility of device outages (the device is not repolled within the required time interval) due to the execution of an LPDA test.

---

[11] There can be more than one device if you are specifying a multipoint line.

Multipoint Lines: In rare cases, a succession of tests on consecutive devices can lead to device outages unless larger device time outs are used. To be sure that no outages occur because of LPDA tests on multipoint lines, use the following guidelines for choosing time out values:

- For LPDA1 use device time out values of at least $20 \times n$ seconds.

- For LPDA2 use device time out values of at least $30 \times n$ seconds plus n $\times$ the value coded on ENABLTO on the BUILD definition statement.

n is the number of remote devices on the multipoint line.

Do not view these time out values as the amount of time normally required to perform an LPDA test. These time out values are related to the rare situations in which no response is received to a modem test. When the LPDA test returns a response, the test duration time is much shorter.

3746 Model 900 SDLC Lines: The 3746 Model 900 does not support migration mode, so LPDATS=LPDA1 or LPDATS=(LPDA1,3867) is not valid when you specify a relative line number greater than 2048 for ADDRESS on this LINE definition statement.

LPDA2DL

```
         ┌─LPDA2DL=NO────────┐
►►──────┼───────────────────┼──────►◄
         └─LPDA2DL=──┬─YES─┬─┘
                     └─NO──┘
```
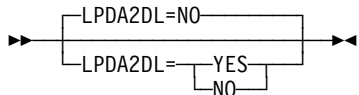
Use: SDLC and BSC lines with externally clocked modems in network control mode
Default: NO

Specifies whether the line uses modems that are not actually LPDA2 modems, but that can support the LPDA2 modem dial facility.

You must define a switched line (DIAL=YES). LPDA2DL=YES is not valid if you code LEVEL2=USER or X21SW=YES on the GROUP definition statement.

These modems support only the LPDA2 dial and disconnect commands. Since these modems are not actually LPDA2 modems, you should omit LPDATS or code LPDATS=NO. If you code a value other than LPDATS=NO, LPDA2DL=YES is not valid.

If you specify LPDA2DL on LINE, you must:

- Code RING=YES
- Code AUTO=YES or not code the AUTO keyword at all

Token Ring: If you specify LPDA2DL, only LPDA2DL=NO is valid.

3746 Model 900 SDLC Lines: The 3746 Model 900 does not support the LPDA2 modem dial facility, so LPDA2DL=YES is not valid when you specify a relative line number greater than 2048 for ADDRESS on this LINE definition statement.

LSPRI

```
                  ┌─LSPRI=PU──────────┐
►►────────────────┴───────────────────┴─►◄
     └─LSPRI=──┬─PU───┬─
               ├─NO───┤
               └─LINK─┘
```
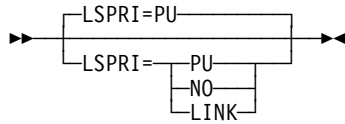
Use: SDLC lines in network control mode; Ethernet-type, Token-Ring,
            ISDN, and frame-relay lines
Default: See text

Specifies whether transmission priority is to be used to determine the order in
which PIUs are transmitted to the physical unit. Do not code LSPRI if you code
VIRTUAL=YES on the GROUP definition statement.

The default is LSPRI=PU for SDLC lines.

PU
> Specifies that outbound data is to be prioritized for each physical unit on the
> line. The priority of sessions across the physical unit is based on the trans-
> mission priority of virtual routes associated with the sessions. LSPRI=PU is
> valid if you specify LNCTL=SDLC for peripheral lines, LNCTL=USER, or
> LEVEL2=*symbol* on the GROUP definition statement.

NO
> Specifies that transmission priority is not to be used as the basis of trans-
> mission order. PIUs are queued in the order they arrive.

LINK
> Replaces the SOT with a prioritized queue used for transmission. The SOT is
> still used for polling. You may code LSPRI=LINK only for type 1 or type 2
> physical units on either duplex or half-duplex lines on which HDXSP=YES is
> specified. LSPRI=LINK is valid only for peripheral SDLC lines or IBM special
> products or user-written code.

> Coding LSPRI=LINK causes transmission priority to become the transmission
> order basis not only across the physical unit but for *all* PIUs across the link.

Token Ring: For Token-Ring physical lines, if you code
ECLTYPE=(PHYSICAL,PERIPHERAL) or ECLTYPE=(PHYSICAL,ANY), you
can code LSPRI=NO or LSPRI=PU. If you code
ECLTYPE=(PHYSICAL,SUBAREA), only LSPRI=NO is valid. For Token-
Ring logical lines, only LSPRI=NO is valid.

Frame Relay: For frame-relay physical lines, if you specify
FRELAY=(PHYSICAL,PERIPHERAL) or FRELAY=(PHYSICAL,ANY), you
can code LSPRI=NO or LSPRI=PU. If you specify
FRELAY=(PHYSICAL,SUBAREA), only LSPRI=NO is valid. For frame-relay
physical lines, the default is LSPRI=NO.

ISDN: For ISDN physical lines, you can specify either LSPRI=NO (the default) or
LSPRI=PU.

Table 2-66 on page 2-245 shows the effect on LSPRI when you code certain
keywords on the GROUP definition statement.

Table 2-66. NDF Values for LSPRI on the LINE Definition Statement

When You Code This on GROUP	The Effect on LSPRI Is
LEVEL2	The default is LSPRI=NO.
ETHERNET=PHYSICAL	The default is LSPRI=NO. The only valid value is LSPRI=NO.
FRELAY=PHYSICAL and ADDRESS=0–895 FRELAY=(PHYSICAL,SUBAREA)	The default is LSPRI=NO. The only valid value is LSPRI=NO.
FRELAY=PHYSICAL and ADDRESS=1024–1039 FRELAY=(PHYSICAL,PERIPHERAL) FRELAY=(PHYSICAL,ANY)	The default is LSPRI=NO.
FRELAY=PHYSICAL or FRELAY=(PHYSICAL,ANY) and ADDRESS=2112-2750	The default is LSPRI=NO.
ECLTYPE=PHYSICAL ADDRESS=2112–2750	The default is LSPRI=NO.
ISDN=PHYSICAL	The default is LSPRI=NO.

MAXDLCI

```
►►─┬──────────────────────────────────┬─►◄
   └─MAXDLCI=(──address──,─┬─D──┬─)─┘
                           ├─H──┤
                           └─DD─┘
```
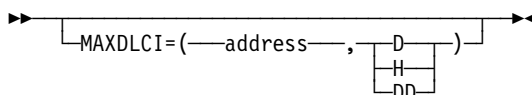
Use: Frame-relay physical lines; NTRI frame handler logical links; ISDN physical lines
Default: See text
Minimum: X'10'
| Maximum: See Table 2-67 on page 2-246

Frame Relay: MAXDLCI specifies the largest DLCI that can be used on this
| frame-relay interface. The default is the maximum allowed value;
| see Table 2-67 on page 2-246.

ISDN: MAXDLCI specifies the largest DLCI used by all associated
                  B-channel lines thereby limiting the DLCI range supported by the
                  B-channels.

                  Note: There is no default value for ISDN MAXDLCI.

Specifying a MAXDLCI value reduces the number of DLCIs reported in LMI status
messages sent to the adjacent mode. MAXDLCI can be specified in three number
bases:

- Decimal
- Hexadecimal
- Dotted decimal

If only a DLCI value is specified (such as MAXDLCI=254), then a decimal value is
assumed.

MAXDLCI definition format and definition rules are as follows:

- MAXDLCI=(dlci,D) — D means MAXDLCI is specified in decimal representation)
- MAXDLCI=(dlci,H) — H means MAXDLCI is specified in hexadecimal representation)
- MAXDLCI=(dlci,DD) — DD means MAXDLCI is specified in dotted decimal representation)

If the LMI protocol is not supported for this line (LMI=NO on the first PU definition statement for this line), the maximum value for MAXDLCI is 991.

If an LMI protocol is supported for this line (LMI is coded and is not equal to NO on the first PU definition statement for this line), the maximum value for MAXDLCI is determined by the following expression:

$$((maxframe - n) / 5) + 15$$

Where:

maxframe is the value of MAXFRAME for this line.

n is 17 when LMI=CCITT or LMI=CCITT-NUI on the LMI PU definition statement.

n is 18 when LMI=ANSI or LMI=ANSI-NUI on the LMI PU definition statement.

The range of values for MAXDLCI is shown in Table 2-67.

Table 2-67. Values for MAXDLCI on the LINE Definition Statement		
LMI=NO	**LMI=CCITT** **LMI=CCITT-NUI**	**LMI=ANSI** **LMI=ANSI-NUI**
16 to 991	16 to ((MAXFRAME-17)/5) + 15 with an upper limit of 991	16 to ((MAXFRAME-18)/5) + 15 with an upper limit of 991

The range depends on the LMI support specified and the line's MAXFRAME value.

MAXFRAME

```
         ┌─MAXFRAME=2106─┐
►►───────┤               ├───►◄
         └─MAXFRAME=size─┘
```
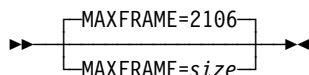
Use:	Frame-relay and ISDN physical lines NTRI frame handler logical lines
Default:	2106
Minimum:	282
Maximum:	8250

Specifies the maximum size, in bytes, for frames transmitted over this physical or logical line.

See *NCP and EP Reference* for more information on how frame size affects multi-link transmission group performance.

Note: The value of MAXDATA on all PU definition statements under this LINE definition statement must not exceed the value of MAXFRAME.

Multilink Transmission Groups: The value of MAXFRAME can affect the segment size of a MLTG that contains this line. For more information, refer to the discussion of MLTG segment size in the *NCP, SSP, and EP Resource Definition Guide*.

Frame Relay: Code the same MAXFRAME value for all physical lines that support segments of a PVC.

For connections using the frame-relay BNN format, this frame size includes the preceding and trailing flags (2 bytes), the DLCI (2 bytes), the RFC1490 header (6 bytes), the logical link control (LLC) header (4 bytes), and the frame check sequence (FCS) (2 bytes).

For connections using the frame-relay BAN format, this frame size includes the preceding and trailing flags (2 bytes), the DLCI (2 bytes), the RFC1490 802.5 bridge header (10 bytes), the MAC destination and source addresses (12 bytes), Routing Information field (up to 18 bytes), the LLC header (4 bytes), the PIU, the optional LAN FCS (4 bytes), and the frame relay FCS (2 bytes).

If this line supports the LMI protocol, as specified by the LMI keyword on the first PU definition statement for this line, the value of MAXFRAME might affect the maximum value of DLCI (or ADDR) on the PU definition statements for this line. For more information, see DLCI on page 2-377.

NTRI: MAXFRAME must be coded on the physical line when a frame-relay physical unit is defined on the line. You can also code MAXFRAME on the NTRI frame handler logical line.

Note: MAXFRAME does not affect the maximum frame size of SNA and IP data sent and received over the TIC. These protocols are controlled by MAXTSL. See MAXTSL on the LINE definition statement on page 2-249 for more information about MAXTSL values.

Physical Lines: Code a MAXFRAME value on the physical line that is greater than or equal to the MAXFRAME value coded on any corresponding NTRI frame handler logical line.

The larger value of MAXFRAME and MAXTSL specified on the physical line, determines the maximum frame-relay frame size transmitted on the line. If MAXFRAME is the larger value, it must not exceed maximum MAXTSL value because frame-relay frames transmitted over a NTRI physical line are not segmented. In addition, any bridges between adjacent NCPs should support MAXFRAME bytes of data or activation of the NTRI frame handler logical line may fail. The frame-relay physical lines flags (2 bytes) and FCS (2 bytes) is replaced by 4 bytes of 802.2 LLC for NTRI frame handler frames in the MAXFRAME frame size.

Note: The maximum transmission frame size is compared to RCVBUFC to determine the maximum frame size that can be received by the physical line. For more information on frame sizes, refer to Chapter 6, "Performance Characteristics", in the *NCP, SSP, and EP Resource Definition Guide*.

Frame Handler Logical Lines: A valid MAXFRAME value coded on the logical line determines the maximum frame-relay frame size for that logical line. For remote connections, NCP will fail activation of the NTRI frame

handler logical line if the MAXFRAME value is not supported by Token-Ring bridges between adjacent NCPs.

If this line supports the LMI protocol, as specified by the LMI keyword on the first PU definition statement for this line, the value of MAXFRAME might affect the maximum value of DLCI (or ADDR) on the PU definition statements for this line. For more information, see DLCI on page 2-377.

MAXPU

```
►►──┬─────────────────┬──►◄
    └─MAXPU=count─┘
```
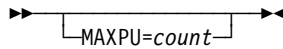
Use: SDLC, frame-relay, Token-Ring, and Ethernet-type lines; ESCON logical links
Default: See text
Minimum: Number of PU definition statements following this LINE definition statement
Maximum: See text

Specifies the maximum number of physical units that can be associated with the link. If this is a switched link, MAXPU must equal 1. If dynamic reconfiguration is not supported and you code MAXPU, *count* must equal the number of physical units defined for this line.

Physical units can be added or moved to a nonswitched SDLC link if (1) dynamic reconfiguration is supported, (2) *count* is greater than the number of PU definition statements, and (3) the SOT contains an extension. The SOT extension is defined within MAXLIST on the SERVICE definition statement. Be sure the number of entries in this extension is equal to or greater than the number of physical units that can be added to the link.

MAXPU is not valid if you code VIRTUAL=YES on the GROUP definition statement for this line.

Unless otherwise stated in Table 2-68 on page 2-249, the default is the number of PU definition statements following this LINE definition statement.

To help determine the value of MAXPU, use the following formula:

The value of MAXPU minus the number of physical units generated must be less than or equal to the value coded for MAXLIST on the SERVICE definition statement minus either (1) the number of entries coded for ORDER on the SERVICE definition statement or (2) if ORDER is not defined, the number of PU resources for the line.

For casual connection, code MAXPU=1 if you specify MODE=SECONDARY on the GROUP definition statement.

ESCON Links: For ESCON logical links, the maximum value for MAXPU is 16. MAXPU is not valid for ESCON physical links.

NTRI Physical Links: MAXPU must equal the number of PUs associated with the physical line. The Token-Ring physical line can support an IP, a SNA, and a frame-relay physical unit. The maximum value for MAXPU, and the default value, is the number of physical PUs defined.

NTRI Frame Handler Logical Lines: MAXPU must be large enough to support the LMI station and any FHSP stations defined or dynamically reconfigured on the line. The maximum value for MAXPU, and the default value, is 976. The minimum value is 2.

3746 Model 900 Token Ring: MAXPU must equal the number of PUs associated with the physical line. The Token-Ring physical line can support one SNA physical unit and one IP physical unit. The maximum value for MAXPU is 2, and the default value is the number of physical PUs defined.

Frame-Relay Physical Lines: For nonswitched lines, the maximum value for MAXPU, and the default value, is 976. For switched lines (DIAL=YES on the GROUP statement), MAXPU=1 is the only valid value.

Table 2-68 shows the effect on MAXPU when you code certain keywords on the GROUP definition statement.

Table 2-68. NDF Values for MAXPU on the LINE Definition Statement

When You Code This on GROUP	The Effect on MAXPU Is
ECLTYPE=PHYSICAL	The only valid value is MAXPU=1, MAXPU=2, or MAXPU=3.
ECLTYPE=LOGICAL,FRELAY	The default is MAXPU=976. The minimum is 2.
FRELAY=LOGICAL	The only valid value is MAXPU=1.
FRELAY=PHYSICAL and DIAL=NO	The default is MAXPU=976.
FRELAY=PHYSICAL and DIAL=YES	The only valid value is MAXPU=1.
ETHERNET=PHYSICAL	The default is MAXPU=1. The only valid value is MAXPU=1.

MAXTSL
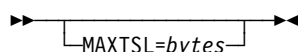
▶▶────────────────────────▶◀
    └─MAXTSL=*bytes*─┘

Use: Token-Ring physical lines
Default: See text
Minimum: See text
Maximum: See text

Specifies the maximum amount of data, in bytes, including the transmission header and the request/response header, that can be transmitted over the Token-Ring connection. MAXTSL is valid only when you code physical resources for ECLTYPE on the GROUP definition statement. See the *NCP and EP Reference* for more information on how the amount of data to be transmitted affects multilink transmission group performance.

See MAXFRAME on the LINE definition statement on page 2-246 for related information about NTRI frame handler support.

Table 2-69 lists the default, minimum, and maximum values for MAXTSL.

Table 2-69. Values for MAXTSL on the LINE Definition Statement

| TIC Type | ECLTYPE= | MAXTSL Values | | |
		Default	Minimum	Maximum
TIC1	(PHYSICAL,PERIPHERAL)	692	265	2044
TIC1	(PHYSICAL,SUBAREA)	692	266	2044
TIC2 or TIC3	(PHYSICAL,PERIPHERAL) with TRSPEED=4	2012	265	4060
TIC2 or TIC3	(PHYSICAL,PERIPHERAL) with TRSPEED=16	2012	265	16732
TIC2 or TIC3	(PHYSICAL,SUBAREA) or (PHYSICAL,ANY) with TRSPEED=4	2012	266	4060
TIC2 or TIC3	(PHYSICAL,SUBAREA) or (PHYSICAL,ANY) with TRSPEED=16	2012	266	16732

Multilink Transmission Groups: The value of MAXTSL can affect the segment size of a MLTG that contains this line. For more information, refer to the discussion of MLTG segment size in the *NCP, SSP, and EP Resource Definition Guide*.

MLTGPRI

```
          ┌─MLTGPRI=0────────┐
►►────────┼──────────────────┼──────►◄
          └─MLTGPRI=priority─┘
```
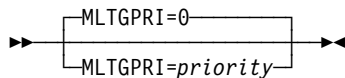
Default: 0
Minimum: 0
Maximum: 255

Assigns a priority to a line in a MLTG. The lowest priority is 0 and the highest priority is 255. NCP selects higher priority lines before lower priority lines when you code MLTGORDR=MLTGPRI on the BUILD definition statement. If you code MLTGORDR=FIFO on the BUILD definition statement or if you do not code MLTGORDR, the MLTGPRI keyword is ignored. See MLTGORDR on the BUILD definition statement on page 2-38 for more information.

For switched PUs, MLTGPRI can be coded in the VTAM switched major node with the appropriate level of VTAM. Values coded for MLTGPRI in the VTAM switched major node will override the values coded or taken as default in the NCP definition.

MODEM

```
          ┌─MODEM=OPTION2──────┐
►►────────┼────────────────────┼──────►◄
          └─MODEM=─┬─OPTION2─┬──┘
                   ├─OPTION1─┤
                   └─NTT─────┘
```
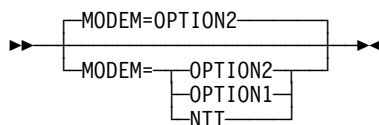
Use: Start-stop and BSC lines in emulation mode
Default: OPTION2

Specifies whether this line is to be enabled or disabled after the communication controller is loaded (by IPL) or the System Reset key is pressed.

OPTION2
Specifies that this line is to be enabled immediately after the controller is loaded or the System Reset key is pressed.

OPTION1
Specifies that this line is to be disabled immediately after the controller is loaded or the System Reset key is pressed, and subsequently must be enabled by command from the host. MODEM=OPTION1 is valid only for lines attached to IBM 1030, 1050, 1060, 2740 Model 1 or 2, 2741, or System/7 devices.

NTT
Specifies that the modems on the line are from the Nippon Telegraph and Telephone Company.

Until the line is enabled, most commands issued to the line by the host result in an intervention-required indication to the host. Which procedure is appropriate depends on the data-set-ready lead in the modem that attaches the line to the controller.

If the data-set-ready lead of the modem is continuously energized, specify MODEM=OPTION1. If the data-set-ready lead is *not* continuously energized (and the host must issue an enable command to enable the line), specify MODEM=OPTION2 or omit modem.

EP Standalone: Enable and disable commands are not applicable to lines to which IBM 2845 or 2848 Control Units are attached. Therefore, such lines are always enabled when the controller is loaded or the system reset key is pressed; it is necessary to specify MODEM=OPTION1.

MONITOR

```
                 ┌─MONITOR=NO─┐
►►──────┼────────────────┼──────►◄
                 └─MONITOR=─┬─YES─┬─┘
                           └─NO──┘
```
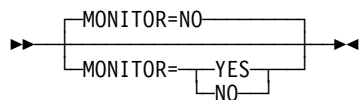
Use: IBM 1050 and IBM 2741; TWX and multiple-terminal-access lines in network control mode

Default: NO

Specifies whether NCP is to monitor this line for an attention signal or for a disconnect condition at the terminal. If MONITOR=YES, NCP monitors the line to determine whether it is executing a command for that line.

The access method is notified when either an attention signal or a disconnect condition is detected.

If you code the logical keyboard lock for TWX terminals (KBDLOCK on the GROUP definition statement), specify MONITOR=YES.

MONLINK

```
►►──┬────────────────────────────────┬──►◄
     └─MONLINK=─┬─YES────────┬─────────┘
               ├─NO─────────┤
               └─CONTinuous─┘
```
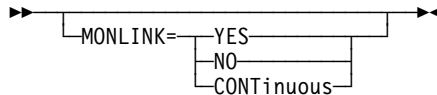
Use: SDLC subarea links, channel links, ESCON logical links, frame-relay, ISDN, and Token-Ring logical subarea lines

Default: See text

Specifies whether the SSCP monitor mode function (SMMF) is to initially activate the link. If you code MONLINK=YES or MONLINK=CONTINUOUS, SMMF will activate the link. If you code MONLINK=NO, VTAM must activate the link.

MONLINK is valid only when PUTYPE=4 or PUTYPE=5 is specified on the PU definition statement; see the note about Multipoint Subarea Links on page 2-252. MONLINK is not valid when VIRTUAL=YES is specified on the GROUP definition statement.

YES
> Specifies that SMMF is to activate the link.
>
> For enhanced loading and activation of remote controllers, MONLINK=YES is valid for switched subarea (SDLC, X.25, and X.21) links and for Token-Ring logical subarea lines.

NO
> Specifies that SMMF is not to manage the link; VTAM must activate the link.

CONTinuous
> Specifies that SMMF is to monitor the link continuously. If you specify MONLINK=CONTINUOUS, SMMF does not release the link when an external SSCP becomes the owner of the NCP. If you specify MONLINK=CONTINUOUS on at least one line, the NCP can have up to seven other owning SSCPs. SMMF will always be one of the owners of the NCP. The value of the MAXSSCP keyword on the BUILD definition statement includes SMMF.
>
> You can code MONLINK=CONTINUOUS for nonswitched subarea lines or channel-attached lines with PUTYPE=5 or for switched subarea lines.

Lines Attached to a 3745 Low-Speed Scanner: The scanner scans only LICs with active lines, to increase efficiency. Coding MONLNK=YES can cause a line to become active, thus increasing the number of lines that the scanner must scan. Care must be taken not to overload the scanner. For information about scanner capacities, see *3745 Communication Controller, All Models: Connection and Integration Guide*, SA33-0129.

Switched Subarea Links: If you code MONLINK=YES or MONLINK=CONTINUOUS on a switched subarea line, also code the SWMMTID keyword on this LINE definition statement. Only incoming calls are supported; SMMF cannot initiate a call.

Multipoint Subarea Links: If you define this NCP as the primary link station on a multipoint subarea link (that is, if you code MODE=PRIMARY on the GROUP definition statement and ADDR on the PU definition statement and do not code SDLCST on the LINE definition statement), the value you

specify for MONLINK applies only to subarea nodes (PUTYPE=4 or PUTYPE=5) attached to the link.

ESCON Links: MONLINK=CONTINUOUS is the default for ESCON logical links. If you code MONLINK=YES or MONLINK=CONTINUOUS for an ESCON logical link, also code XMONLNK=YES on the PU definition statement for the physical link station.

Specify MONLINK=NO if you have only peripheral stations. If you specify MONLINK=YES or MONLINK=CONTINUOUS and this line has both peripheral and subarea stations, SMMF supports only the subarea stations.

To load or dump NCP over an ESCON link, do the following:

- Code MONLINK=YES or MONLINK=CONTINUOUS on the LINE definition statement for the logical line.

- Code MONLINK=YES on the PU definition statement for the logical link station you want to use.

- Code XMONLNK=YES on the PU definition statement for the physical link station.

Token Ring, Frame Relay, and ISDN: If you code MONLINK=YES or MONLINK=CONTINUOUS for a logical subarea line, also code XMONLNK=YES on the PU definition statement that defines the link station for the associated physical line. For ISDN, you also need to code XMONLNK=YES on at least one ISDN B-channel PU definition statement.

Table 2-70 shows the effect on MONLINK when you code certain keywords on the BUILD, GROUP, and PU definition statements.

Table 2-70. NDF Values for MONLINK on the LINE Definition Statement

When You Code This	The MONLINK Default Is
BUILD: TYPGEN=NCP or TYPGEN=PEP	NO
BUILD: TYPGEN=NCP-R	YES
GROUP: LNCTL=CA and PU: PUTYPE=5	CONTINUOUS

MPTALT

```
        ┌─MPTALT=NO─┐
►►──────┤           ├──────►◄
        └─MPTALT=─┬─YES─┬─┘
                  └─NO──┘
```
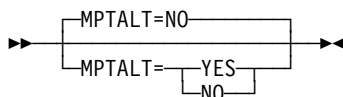
Use: Switched callout BSC line in network control mode
Default: NO

Specifies whether a switched BSC line can be used as an alternate line for communicating with BSC stations normally reached over a nonswitched multipoint line. When failure of the multipoint line prevents NCP from communicating with a station over that line, the switched network backup facility can be used to reach the station over any switched callout BSC line that has MPTALT=YES specified.

MPTALT is valid only if the line is a switched callout BSC line (code CALL=OUT or CALL=INOUT on this LINE definition statement). If you specify these keywords and omit MPTALT, NO is the default.

MTALIST

```
►►─┬──────────────┬─►◄
   └─MTALIST=name─┘
```
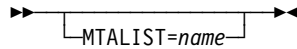
Use: Call-in MTA lines in network control mode
Default: None

Specifies that this line will be used as a call-in multiple-terminal-access line and identifies the list NCP is to use to identify the type of terminal calling the controller over this line. *name* is the name of an MTALIST definition statement that defines the list that identifies the terminal type.

MTALIST is valid only if you code DIAL=YES and LNCTL=SS on the GROUP definition statement, and CALL=IN or CALL=INOUT on this LINE definition statement.

MULTI

```
      ┌─MULTI=NO─────┐
►►─────┼──────────────┼────►◄
      └─MULTI=─┬─YES─┬┘
              └─NO──┘
```
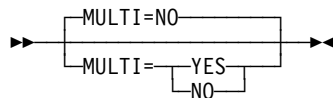
For: IBM 2845 and 2848 control units in emulation mode
Default: NO

Specifies that the line represented by this LINE definition statement connects the controller to (1) one or more IBM 2845 or 2848 control units equipped with multi-point line control (MULTI=YES), or (2) a single display control equipped with point-to-point line control (MULTI=NO).

NCPCA

```
      ┌─NCPCA=ACTIVE───────┐
►►─────┼────────────────────┼────►◄
      └─NCPCA=─┬─ACTIVE───┬┘
              └─INACTIVE─┘
```
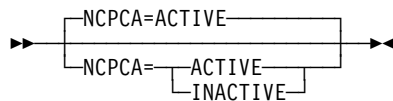
Use: Channel links
Default: ACTIVE for NCP and PEP
           INACTIVE for EPSA

Specifies the active or inactive status of the channel adapter. If NCPCA=ACTIVE, NCP or EP uses the native subchannel. If NCPCA=INACTIVE, NCP or EP does not use the native subchannel, and HICHAN and LOCHAN must be specified on LINE.

You may not code NCPCA=INACTIVE when you code CA=TYPE7 or CA=TYPE7-TPS.

ESCON Links: The NCPCA keyword is not valid.

NEGPOLP

```
              ┌─NEGPOLP=NONE─────────────┐
▶▶──────┬─────────────────────────┬──────▶◀
        └─NEGPOLP=─┬─seconds─┐
                   └─NONE────┘
```
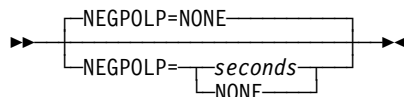
Use: Multipoint BSC lines in network control mode
Default: NONE
Minimum: 0.1
Maximum: 23.5

Specifies how long NCP is to pause before resuming polling after receiving a negative response to polling from a multipoint BSC station. The pause can be defined in seconds or 10ths of seconds.

If you specify NEGPOLP=NONE, there is no pause between receipt of the negative response and continuation of polling.

A negative polling pause limits the amount of nonproductive polling on a BSC multipoint line and thereby reduces the amount of processing overhead. However, too large a negative polling pause can increase the response time for terminals on this line.

NEWSYNC

```
              ┌─NEWSYNC=NO────┐
▶▶──────┬─────────────────────┬──────▶◀
        └─NEWSYNC=─┬─YES─┐
                   └─NO──┘
```
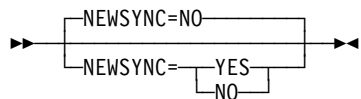
Use: BSC and SDLC lines (NCP or EP)
Default: See text

Specifies whether the communication controller supplies the new-sync signal to the modem (data set) on this line.

NEWSYNC=YES is valid only if all the following apply:

* The modem (at the controller) serving the line has the new-sync feature.

* The communication controller is the multipoint primary device for a duplex line.

* The modem at the link-attached device does not continuously send carrier signals to the modem at the controller.

* The modem at the link-attached station can send a continuous carrier signal, or the station can send a continuous request-to-send to the modem

* You are defining nonswitched lines.

Consult your IBM representative or the installer of the modem for further information about these signals.

For an X.21 line, code NEWSYNC=NO or omit this keyword.

Table 2-71 on page 2-256 shows the effect on NEWSYNC when you code certain keywords on the GROUP and LINE definition statements.

Table 2-71. NDF Values for NEWSYNC on the LINE Definition Statement

When You Code This	The Effect on NEWSYNC Is
GROUP: MODE=SECONDARY for casual connection	The only valid value is NEWSYNC=NO.
GROUP: ETHERNET=PHYSICAL	The only valid value is NEWSYNC=NO.
GROUP: DIAL=NO and LINE: POLLED=YES (BSC only), DUPLEX=FULL, and CLOCKNG=EXT	The default is NEWSYNC=YES.

NPACOLL
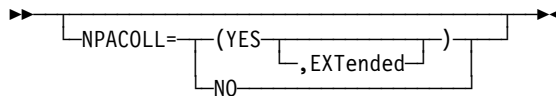
Default: See text

Specifies whether performance data for this line or channel is to be collected by the network performance analyzer.

NO|YES

Specifies whether performance data is to be collected. NPACOLL=YES is valid only if you specify all of the following:

- NPA=YES on the BUILD definition statement

- One of the following on the GROUP definition statement:

      LNCTL=SDLC
      LNCTL=BSC
      LNCTL=CA
      ECLTYPE=PHYSICAL
      ECLTYPE=LOGICAL
      ETHERNET=PHYSICAL
      FRELAY=PHYSICAL
      FRELAY=LOGICAL
      ISDN=(LOGICAL,B)
      ISDN=(LOGICAL,SUBAREA)
      ISDN=(LOGICAL,PERIPHERAL)

NPACOLL=YES is not valid if LNCTL=BSC and any terminal or cluster controller other than an IBM 3271 or 3275 is defined on the line.

EXTended

Indicates that performance data collected by the network performance analyzer is to be forwarded to NPM using extended format COLLECT request units (RUs).

If you code ECLTYPE=PHYSICAL, ETHERNET, FRELAY, or ISDN on the GROUP definition statement or if you are defining a 3746 Model 900 SDLC line or an ESCON link, and if you code NPACOLL=YES on the LINE or GROUP definition statement, NDF uses NPACOLL=(YES,EXTENDED).

IP over Switched Frame-Relay Physical Lines: NPACOLL=YES is valid on the physical LINE statement and on the logical PU statement, but **not** on the logical LINE statement.

3746 Model 900 Token Ring: For 3746 Model 900 Token-Ring logical lines, NPACOLL=NO is the only valid value. If NPACOLL=(YES,EXTENDED) on the GROUP definition statement and you do not code NPACOLL on the LINE definition statement, NDF adds NPACOLL=NO on the LINE definition statement.

Table 2-72 shows the effect on NPACOLL when you code NPA=YES on the BUILD definition statement and certain keywords on the GROUP and LINE definition statements.

Table 2-72. NDF Values for NPACOLL on the LINE Definition Statement

When You Code This	The Effect on NPACOLL Is
GROUP: LNCTL=CA	The default is NPACOLL=YES.
ESCON resources	The default is NPACOLL=(YES,EXTENDED). (See description of EXTENDED.)
GROUP: LNCTL=SDLC and VIRTUAL=NO LINE: ADDRESS<2048	The default is NPACOLL=YES.
GROUP: LNCTL=SDLC and VIRTUAL=NO LINE: ADDRESS≥2048	The default is NPACOLL=(YES,EXTENDED).
GROUP: VIRTUAL=YES or LNCTL=BSC	The default is NPACOLL=NO.
GROUP: ECLTYPE=LOGICAL	The default is NPACOLL=NO. If NPACOLL=YES, then NDF adds NPACOLL=(YES,EXTENDED). (see note for 3746 Model 900 Token Ring)
GROUP: ECLTYPE=PHYSICAL	The default is NPACOLL=NO. (See description of EXTENDED.)
GROUP: ETHERNET	The default is NPACOLL=NO. (See description of EXTENDED.)
GROUP: FRELAY	NDF adds NPACOLL=(YES,EXTENDED). (See description of EXTENDED.)
GROUP: ISDN	NDF adds NPACOLL=(YES,EXTENDED). (See description of EXTENDED.)

NPATP
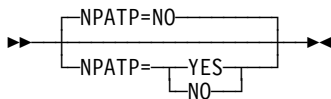
```
         ┌─NPATP=NO─────┐
►►───────┤              ├──────►◄
         └─NPATP=─┬─YES─┘
                  └─NO──┘
```

Use: SDLC resources
Default: NO

Specifies whether you want network performance data to be collected by PIU transmission priority on the line in addition to basic network performance data. Code NPATP=YES if you want network performance data to be collected by transmission priority. If you code NPATP=YES, you must also code NPACOLL=YES for this resource.

NPATP is not valid if you code ECLTYPE, ETHERNET, FRELAY, LEVEL2, NCST, or VIRTUAL=YES on the GROUP definition statement.

NRZI

```
►►──┬───────────────────┬──►◄
     └─NRZI=─┬─YES─┬─────┘
             └─NO──┘
```
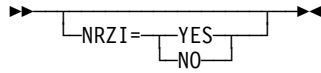
Use: SDLC and frame-relay lines in network control mode
Default: See text

Specifies whether the data terminal equipment at the end of the link must operate in non-return-to-zero change-on-ones (NRZI) mode (NRZI=YES) or in non-return-to-zero (NRZ) mode (NRZI=NO).

If internal clocking is used on the link, specify NRZI=YES or omit the keyword.

If external (modem) clocking is used on the link, specify NRZI=YES or omit the keyword unless the modem is sensitive to repeated binary one-zero patterns (that is, 10101010...). Sensitivity to this pattern can cause the modem to lose synchronism when it encounters sufficiently long sequences of this pattern in message data. If the modem is sensitive to this bit pattern, specify NRZI=NO. To determine whether your modems are sensitive to repeated binary one-zero patterns, consult your IBM representative (for IBM modems) or the supplier or installer of the modem (for non-IBM modems).

Code NRZI=NO for most non-IBM modems and all IBM 5812 and 5822 data set units. Code NRZI=YES for IBM modems.

NRZI is valid only if you define LNCTL=SDLC on the GROUP definition statement.

Frame Relay: The default is NRZI=NO.

SDLC Links: The default is NRZI=YES.

Attention: All terminal equipment on the same link must use the same encoding scheme; that is, they all must use NRZI mode (NRZI=YES) or all must use NRZ mode (NRZI=NO). Mixing of modes on the same link results in total lack of communication between stations on the link. In the case of an link between communication controllers, the equivalent choice (NRZI=YES or NRZI=NO) is required for the LINE definition statement in each program that represents the link, and must correspond to the setting of the NRZI bit in the IPL configuration data set of the remote program loader (when the controller is link-attached). (The IBM customer engineer sets the NRZI bit in the communication controller.) Specifying NRZI or NRZ is part of the customization process for some terminals.

Table 2-73 shows the effect on NRZI when you code certain keywords on the GROUP definition statement.

Table 2-73. NDF Values for NRZI on the LINE Definition Statement

When You Code This on GROUP	The Effect on NRZI Is
ETHERNET=PHYSICAL	The default is NRZI=NO. The only valid value is NRZI=NO.
FRELAY=PHYSICAL	The default is NRZI=NO.

PAD
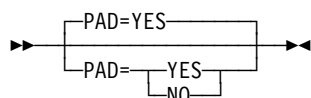
```
                   ┌─PAD=YES─────┐
►►─────────────────┴─────────────┴──►◄
          └─PAD=──┬─YES─┬──┘
                  └─NO──┘
```

Use: BSC lines in emulation mode
Default: YES

Specifies whether NCP, when emulating IBM 2703 Transmission Control, verifies the first 4 bits of trailing pad characters received from the lines.

If you specify PAD=YES or omit the keyword, the controller checks each pad character received and indicates a data check error if the first 4 bits are not all ones. If you specify PAD=NO, the controller does not check the pad characters.

PARCHK

```
              ┌─PARCHK=NOCHECK───────────┐
►►────────────┴──────────────────────────┴──►◄
      └─PARCHK=──┬─EVEN────┬──┘
                 ├─MARK────┤
                 ├─NOCHECK─┤
                 ├─ODD─────┤
                 └─SPACE───┘
```
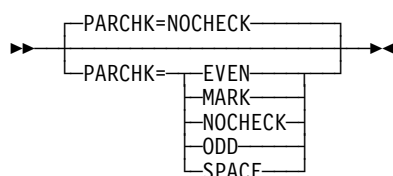
Use: TWX lines in network control mode
Default: NOCHECK

Specifies whether NCP is to check for odd, even, mark, or space parity or is not to check parity for TWX terminals on the line. (Refer also to PECHAR on the GROUP definition statement.) For MARK parity, a character has a parity error if the parity bit is not 1. For SPACE parity, a character has a parity error if the parity bit is not 0.

PARGEN

```
              ┌─PARGEN=EVEN────────────┐
►►────────────┴────────────────────────┴──►◄
      └─PARGEN=──┬─EVEN───┬──┘
                 ├─MARK───┤
                 ├─NOGEN──┤
                 ├─ODD────┤
                 └─SPACE──┘
```

Use: TWX lines in network control mode
Default: EVEN

Specifies whether NCP is to set odd, even, mark, or space parity or is not to set parity (leave the parity bit as it is in the translate table) for data transmitted on the line. MARK parity generation always sets the parity to 1, whereas SPACE sets it to 0.

PAUSE

```
                  ┌─PAUSE=(0.2,2.8)─────────────────┐
►►────────────────┼─────────────────────────────────┼────►◄
                  │          ┌─0.2─┐   ┌─2.8─┐        │
                  └─PAUSE=(──┼─────┼─,─┼─────┼──)─────┘
                             └─t1──┘   └─t2──┘
```
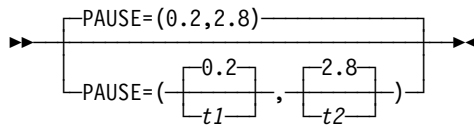
Examples: PAUSE=(0.4,2.0)
             PAUSE=(,3.2)

Specifies the delay or cycle time interval according to the link modulo value.

t1

> Default: 0.2
> Minimum: See text
> Maximum: See text
>
> Specifies the pause for this line if the line is operating as a modulo-8 line. See the following text for more information on this suboperand for different line control types.

t2

> Default: 2.8
> Minimum: See text
> Maximum: See text
>
> Specifies the pause for this line if the line is operating as a modulo-128 line. See the following text for more information on this suboperand for different line control types.

If you specify the SDLCST keyword for this line, the PAUSE value from the SDLCST definition statement is used. To define different values for the primary and secondary state, code the PAUSE keyword on the SDLCST definition statement, not on the LINE definition statement. If no value is provided for PAUSE on the SDLCST definition statement, the value coded for PAUSE on the LINE definition statement is used.

When you generate a subarea link, the value of the REPLYTO keyword in the primary NCP should be greater than the value of the PAUSE keyword in the secondary NCP.

BSC and Start-Stop Lines: Specifies the number of seconds of delay between successive service cycles when no sessions currently exist. *tn* can range from 0 to 255. Define integer (not fractional) values for *tn*. For example, you can specify *tn*=225, but you cannot specify *tn*=225.7. If you omit PAUSE, 0 is the default, and no delay exists between successive cycles. PAUSE applies only to line operation in network control mode and when you define POLLED=YES for this line.

SDLC Links: Specifies the average duration of the polling cycle from 0 to 25.5 seconds. The polling cycle extends from the moment NCP examines the first entry in the SOT to the moment polling next begins at the same entry. The cycle includes time for polling, reading, and writing to the terminals on the line. If the time spent in a complete polling cycle (servicing all active entries in the SOT) equals or exceeds the time specified as *tn*, the next polling cycle begins immediately. If the time spent in a complete polling

cycle is less than *tn*, the beginning of the next polling cycle is deferred until *tn* seconds have elapsed since the beginning of the cycle just completed. During the pause, the SDLC link is in a poll-wait state. Any outgoing data ready for transmission to the SDLC stations on the link is sent during this pause. NCP uses the pause values coded in T1 and T2 (modulo 8, modulo 128) if the line is point-to-point (one entry in the SOT). NCP always uses the value for T1 if the line is multipoint (more than one entry in the SOT).

The larger the number of active entries in the SOT, the more likely it is that the polling cycles will proceed continuously, without intervening pauses.

Allowing a pause to elapse when activity on the link is relatively low can reduce the amount of processing time consumed by unproductive polling.

For large subarea configurations (for example, over 500 lines coded), the PAUSE value for the secondary state must be at least 0.5 second.

Subarea Links: If NCP is in a primary state, specifies the poll pause as described previously. If NCP is in a secondary state, specifies how long NCP can defer returning the SDLC final bit to the adjacent primary NCP.

3746 Model 900 SDLC Links: NCP uses the coded pause value, T1 or T2 (modulo 8, modulo 128) if this is a point-to-point line (one entry in the SOT). NCP uses the value for T1 if this is a multipoint line (more than one entry in the SOT).

POLIMIT
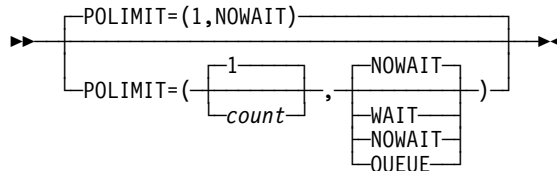
Use: Nonswitched multipoint start-stop or BSC lines in network control mode
Default: (1,NOWAIT)
Examples: POLIMIT=(10,WAIT)
             POLIMIT=(,QUEUE)

Specifies (1) the maximum number of consecutive negative responses NCP is to accept from a polled device, and (2) the action NCP is to take when that number is reached.

POLIMIT applies only to polling performed after NCP has received at least one message block from the device, not to initial polling.

count

    Default: 1
    Minimum: 1
    Maximum: 255

    Specifies the maximum number of consecutive negative responses NCP is to accept from a polled device. When this limit is reached, NCP stops polling the device and starts polling the next device.

The following suboperands specify the action NCP is to perform when the number of negative responses reaches *count*.

WAIT

Specifies that NCP is to (1) maintain the logical connection with the device, (2) notify the host that the negative response limit has been reached, and (3) wait for another request from the host before performing any further action on the line.

NOWAIT

Specifies that NCP is to (1) break the logical connection with the device, (2) notify the host processor, and (3) terminate the current read request.

QUEUE

Specifies that NCP is to (1) break the logical connection with the device, (2) notify the host processor, and (3) queue the current read request onto the beginning of the queue for the device.

VTAM: If NCP runs under VTAM, code the QUEUE suboperand.

IBM 1050 Terminals: For a switched line used to communicate with IBM 1050 terminals, NCP accepts up to 16 consecutive negative responses and then executes the WAIT option.

POLLED

```
                  ┌─POLLED=NO─┐
►►──────────────┼───────────┼──────────►◄
                └─POLLED=─┬─YES─┬─┘
                         └─NO──┘
```
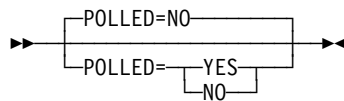
Use: Start-stop and BSC lines in network control mode
Default: NO

Specifies whether start-stop and BSC devices attached to the line must be polled and addressed. POLLED is not valid for SDLC links.

Specify all lines in a line group with the same value (all as POLLED=YES or all as POLLED=NO). You cannot include both options in the same line group.

Code POLLED=YES if:

- The line is a start-stop or BSC multipoint line (except when the controller is a tributary device on a BSC multipoint line).

- The line is a nonswitched or switched point-to-point line (including a multiple-terminal-access line) used to communicate with an IBM 1050 terminal.

Code POLLED=NO if:

- The line is a start-stop nonswitched or switched point-to-point line or a BSC point-to-point line (unless the terminal is an IBM 1050).

- The line is a multiple-terminal-access line (MTALIST keyword of this LINE definition statement is coded) over which no IBM 1050 terminals communicate with the controller.

POLLTO

```
                      ┌─POLLTO=ERROR─────────┐
   ►►─────────────────┴──────────────────────┴────►◄
         └─POLLTO=─┬─ERROR──┬─┘
                   └─NEGRESP─┘
```
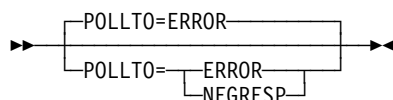
Use: Start-stop lines in network control mode
Default: ERROR

Specifies whether a time out occurring during polling is to be treated as an error condition (POLLTO=ERROR) or a negative response (POLLTO=NEGRESP). If the time out is treated as an error condition, the value you specify for CRETRY on the GROUP definition statement determines the number of subsequent attempts at polling. If the time out is treated as a negative response, the value you specify for POLIMIT on this LINE definition statement determines the number of times the station is repolled.

PORT

```
   ►►────────────────────────►◄
         └─PORT=─┬─A─┬─┘
                 ├─B─┤
                 ├─C─┤
                 └─D─┘
```
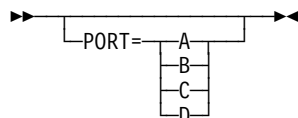
Use: SDLC or BSC lines attached to a local IBM 786x modem with the data multiplexer feature
Default: See text

Indicates which modem port this line is connected to.

Code the PORT keyword only if you code CHNLZ=YES and LPDATS=LPDA2. If you do not code CHANLA, you must code PORT. If you code CHANLA=YES, you may omit PORT. The default is PORT=A if you code CHANLA=YES.

If you code CHANLA=NO, code PORT=B, PORT=C, or PORT=D.

PORTADD

```
   ►►────────────────────────────►◄
         └─PORTADD=number─┘
```
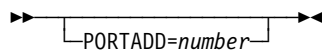
Use: Token-Ring, frame-relay, and ISDN physical lines
           Required for Token-Ring, ISDN, and for 3746 Model 900 frame-relay physical lines
Default: None
Minimum: 0
Maximum: 254 for 3746 Model 900 frame relay and ISDN
           99 for Token Ring and 3745 frame relay

Assigns a port number to this physical line. This number is used to select a physical line during connect-out processing for Token-Ring, frame-relay, and ISDN logical peripheral links. PORTADD is also used during connect-out processing for ISDN logical subarea links.

You can select this line to establish a switched connection by coding DLCADDR=(2,D,*number*) on the VTAM PATH definition statement for the switched

major node definition. You can also select this line by using *number* for the first two digits of the DIALNO value on the VTAM PATH definition statement.

Unless you are performing dial-out processing over duplicate TIC addresses, the value you code for PORTADD must be unique.

Duplicate TIC Addressing: For dial-out processing, code the same PORTADD value, and the same LOCADD value, for all physical lines.

- For NCP V7R4 and earlier releases, the above statement is valid only when you code ECLTYPE=(PHYSICAL,PERIPHERAL) on the GROUP definition statement and neither physical line supports NTRI frame handler logical connections.

- Starting in NCP V7R5, the above statement is valid only when you code ECLTYPE=(PHYSICAL,PERIPHERAL|ANY) on the GROUP definition statement and only one of the physical lines can support NTRI frame handler logical connections.

See "Duplicate TIC Addressing" in the *NCP, SSP, and EP Resource Definition Guide* for more information about duplicate TIC addressing for:

- Logical peripheral connections
- Logical subarea connections
- NTRI frame handler logical connections

NTRI: You can use PORTADD to associate this physical line with a group of logical lines. To do this, code the same value for PORTADD and for PHYPORT on the GROUP definition statement for the group of logical lines. You can also do this using PHYSRSC on the GROUP definition statement. See page 2-130 for more information. You can code the same PORTADD value as other NTRI lines with the same LOCADD value. Otherwise, the PORTADD value must be unique.

3746 Model 900 Token Ring: To associate a physical line with a group of logical lines, code PHYSRSC on the GROUP definition statement. You can code the same PORTADD value as other Token-Ring 3746 Model 900 lines with the same LOCADD value. Otherwise, the PORTADD value must be unique.

Frame-Relay Physical Lines: Code PORTADD for frame-relay physical lines to assign a port number to the physical line. Do not code the same PORTADD value as any other 3745 frame-relay physical line or any Token-Ring line. For 3745 frame relay, you can define up to 155 frame-relay physical lines with no value specified for PORTADD. If you do not specify a value for PORTADD, the frame-relay physical line will not be available for outgoing logical peripheral connections because the port number is not known.

3746 Model 900 Frame Relay: PORTADD is a required keyword. Do not code the same PORTADD value on any other LINE definition statement for a 3746 Model 900 frame-relay line or any Token-Ring line.

ISDN Physical Lines: Code PORTADD for ISDN physical lines to assign a port number to the physical line. PORTADD is a required keyword. Do not code the same PORTADD value as any other ISDN physical line or any Token-Ring line.

PROMPT

```
          ┌─PROMPT=YES─────────┐
►►────────┼────────────────────┼──►◄
          └─PROMPT=──┬─YES─┬────┘
                     └─NO──┘
```
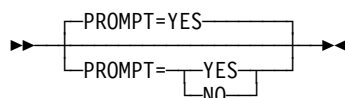
Use: Switched TWX lines in network control mode
Default: YES

Specifies whether, at READ time, NCP is to transmit the PROMPT defined by
TWXID on the BUILD definition statement.

Remember that the TWXID sequence is referred to as an ID at line activation time
and a PROMPT at READ time. Prompting can be overridden by a Set Destination
Mode command and is always bypassed if data has been received in read-ahead
mode.

QUIET

```
          ┌─QUIET=NO──────────┐
►►────────┼───────────────────┼──►◄
          └─QUIET=──┬─YES─┬────┘
                    └─NO──┘
```
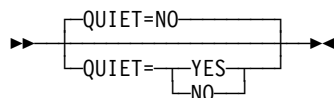
Use: Start-stop lines in emulation mode
Default: NO

Specifies whether the control program observes a long-line quiet time out of 25.6
seconds when receiving from the line. If you code QUIET=YES, NCP or EP
observes the long time out. If you code QUIET=NO (or omit the keyword), NCP or
EP observes the normal time out of 3.0 seconds.

Do not specify QUIET=YES if you code FEATURE=IMEND for this line.

QSIZE
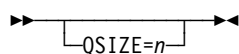
```
►►──────────────────►◄
     └─QSIZE=n─┘
```

Use: 3746 Model 900 frame-relay/ISDN congestion control

Default: Set by the Model 900 based on line speed
Minimum: 0
Maximum: 500

The value of **n** is multiplied by the value of the DATABLK keyword to determine the
threshold in bytes.

Specifies the 3746 Model 900 software queue threshold which, when it is reached,
triggers the discarding of all of the last frames received regardless of how the
Discard Eligible (DE) indicator is set. See the *3745 Communication Controller
Models A, 3746 Expansion Unit Model 900: Migration and Planning Guide*,
GA33-0183, for more information about the Model 900 congestion control proc-
essing.

Note: QSIZE must be more than or equal to QSIZEDE.
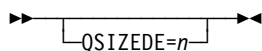
QSIZEDE

```
►►──┬──────────────┬──►◄
     └─QSIZEDE=n─┘
```

Use: 3746 Model 900 frame-relay/ISDN congestion control

Default: Set by the Model 900 based on line speed
Minimum: 0
Maximum: 500

The value of **n** is multiplied by the value of the DATABLK keyword to determine the threshold in bytes.

Specifies the 3746 Model 900 software queue threshold which, when it is reached, triggers the discarding of the last frames received with the Discard Eligible (DE) indicator set on. See the *3745 Communication Controller Models A, 3746 Expansion Unit Model 900: Migration and Planning Guide*, GA33-0183, for more information about the Model 900 congestion control processing.

The value of **n** is multiplied by the value of the DATABLK keyword to determine the threshold in bytes.

Note: QSIZEDE must be less than or equal to QSIZE.

RCVBUFC

```
►►──┬────────────────┬──►◄
     └─RCVBUFC=bytes─┘
```
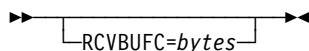
Use: NTRI physical lines
Default: See text
Minimum: Six times the NCP buffer size (BFRS on the BUILD definition statement)
Maximum: See text

Specifies the NTRI buffer capacity for receiving data from a telecommunication link during one data transfer. *bytes* must be greater than or equal to the maximum amount of data that NTRI will receive in one data transfer. For Token-Ring frames containing SNA data, this data includes:

- An 802.5 Token-Ring frame header

- An 802.2 LLC header (4 bytes control and address information)

- SNA PIU (transmission header, request/response header, and request/response unit) data

For Token-Ring frames containing internet data, this data includes:

- An 802.5 Token-Ring frame header

- A Subnetwork Access Protocol (SNAP) header (8 bytes control and address information)

- IP datagram or ARP packet data

For Token-Ring frames containing frame-relay data, this data includes:

- An 802.5 Token-Ring frame header

- An 802.2 LLC header (4 bytes control and address information)

- A frame-relay address header (2 bytes DLCI)

- Frame-relay frame data

Coding the value of RCVBUFC twice the size of the expected amount of data in one data transfer may improve Token-Ring performance.

The default value for RCVBUFC is 4072 when you code ADAPTER=TIC2 and ECLTYPE=(PHYSICAL,SUBAREA) or ECLTYPE=(PHYSICAL,ANY) on the GROUP definition statement; otherwise the default is 1440.

The maximum value for RCVBUFC is 4095 when you code ADAPTER=TIC1 on the GROUP definition statement and 32000 when you code ADAPTER=TIC2.

Multilink Transmission Groups: The value of RCVBUFC can affect the segment size of a MLTG that contains this line. For more information, refer to the discussion of MLTG segment size in the *NCP, SSP, and EP Resource Definition Guide*.

3746 Model 900 Token Ring: The RCVBUFC keyword is ignored for 3746 Model 900 Token-Ring resources.

NTRI Frame Handler: The larger of MAXFRAME and RCVBUFC limits the largest frame-relay frame that can be received on a NTRI physical line.

RCVCHAR

```
           ┌─RCVCHAR=4─────┐
►►─────────┤               ├─►◄
           └─RCVCHAR=─┬─1─┬─┘
                      └─4─┘
```
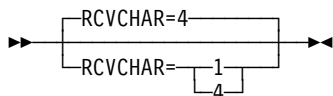
Use: Start-stop lines in emulation mode
Default: 4

Specifies whether a start-stop line receives data in a character mode or in a burst mode. Specify RCVCHAR=1 for character mode or RCVCHAR=4 for burst mode. RCVCHAR is not valid for line speeds greater than 1200 bps.

REDIAL

```
►►─────────────────────────────────────────────────►◄
     └─REDIAL=(─┬───┬─,─┬─0──┬─,─┬─0─┬─,─┬─0──┬─)─┘
                └─m─┘   └─t1─┘   └─n─┘   └─t2─┘
```
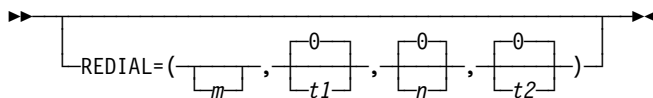
Examples: REDIAL=(5,3,5,6)
                   REDIAL=(,,255,3)

Determines the timing and maximum number of dialing operations NCP performs in attempting to reach a device over a switched line. NCP performs dialing operations in sequences, with a pause between dialing operations in each sequence and a pause between sequences.

m

Default: See text
Minimum: 0
Maximum: 255

Determines the number of dialing operations in each sequence. NCP performs $m + 1$ dialing operations in each sequence.

SDLC Lines: The number of dialing operations in each sequence is provided by the access method, so omit the *m* suboperand.

BSC and Start-Stop Lines: The default value for *m* is 3.

t1

Default: 0
Minimum: 0
Maximum: 765

Specifies, in seconds, the pause between dialing operations in each sequence. You must specify a value that is a multiple of three.

n

Default: 0
Minimum: 0
Maximum: 254

Determines the maximum number of dialing sequences. NCP performs $n + 1$ sequences. Specify 255 for *n* to perform an unlimited number of sequences.

t2

Default: 0
Minimum: 0
Maximum: 765

Specifies, in seconds, the pause between dialing sequences. You must specify a value that is a multiple of three.

REMOTTO
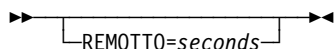
```
►►──────────────────────►◄
     └─REMOTTO=seconds─┘
```

Use: Token-Ring logical subarea resources
Default: Value of REMOTTO on the BUILD definition statement.
Minimum: 0.6
Maximum: 20.0

Specifies the Token-Ring reply timer (T1) value for stations connected to a remote Token Ring. This keyword is valid only if ECLTYPE=(LOGICAL,SUBAREA).

REMOTTO on LINE does not apply to NTRI frame handler logical lines. The value specified on the BUILD statement or its default will be used during test resolve processing for NTRI frame handler logical line connections.

Note: Timer values for logical peripheral lines can be specified on the BUILD definition statement.
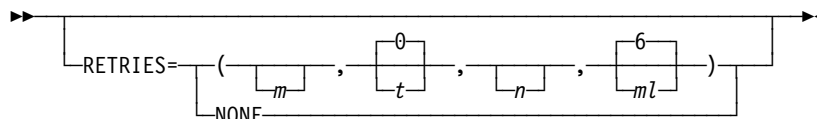
RETRIES

Examples: RETRIES=(5,2,5,)
             RETRIES=(,30,5,120)

Specifies the number of attempts to be made to recover from errors occurring during transmission over SDLC links, BSC and start-stop lines, Token-Ring, frame-relay, and ISDN logical lines. Each attempt at recovery, or *retry*, is a retransmission of data or control commands. NCP performs retries until the transmission is successful or the number of retries reaches the maximum specified by this keyword.

Retries are performed in sets, which are called *retry sequences*. NCP can perform one or more retry sequences. Successive retry sequences are separated by pauses. You can specify the number of retries in a sequence (m), the number of sequences to be performed (n), and the length of the pause between sequences (t). The maximum number of retries NCP attempts equals $(m + 1) \times (n + 1) - 1$.

If you code RETRIES=NONE or RETRIES=0, NCP performs no retries at all.

For 3745 frame-relay physical lines, RETRIES specifies the maximum number of time periods with errors occurring during frame receipt that NCP allows before bringing down the line.

Ethernet-Type LANs: If ETHERNET=PHYSICAL is specified on the GROUP definition statement, only RETRIES=NONE is valid. The default is NONE.

m

Use: SDLC links
            BSC and start-stop lines
            Token-Ring, frame-relay, and ISDN logical lines
Default: See text
Minimum: 0
Maximum: See text

Specifies the maximum number of retries in each retry sequence. For Token-Ring logical lines, m specifies the maximum number of retries for a line established over a remote Token Ring. Table 2-74 on page 2-270 shows the default and maximum values for m for different line control types.

Table 2-74. Values for the m Suboperand of RETRIES on the LINE Definition Statement

Line Control	Default	Maximum
SDLC	15	128
BSC	7	255 [1]
SS	2	255 [1]
Token-Ring, frame-relay, or ISDN logical lines	6	128

[1] 255 specifies unlimited retries.

Use: 3745 frame-relay physical lines
Default: 64 for TSS lines
            254 for HPTSS lines
Minimum: 0
Maximum: 254

Specifies the maximum number of 100 millisecond (ms) periods (during which the FRTE receives at least one frame error) that NCP allows before NCP brings down the physical line and its associated logical, FHSP, and IP resources. Code a value of NONE or 0 to indicate that NCP should **not** bring down the physical line based on consecutive errors.

t

Default: 0
Minimum: 0
Maximum: 255

Specifies the pause interval, in seconds, between successive retry sequences. If you specify 0 for m, also specify 0 for t or omit t.

n

Default: See text
Minimum: See text
Maximum: See text

Specifies the maximum number of retry sequences. If you specify 0 for t, also specify 0 for n or omit n. Table 2-75 shows the default, minimum, and maximum values for n for different line control types.

Table 2-75. Values for the n Suboperand of RETRIES on the LINE Definition Statement

Line Control	Default	Minimum	Maximum
SDLC	0 or 1 [1]	0	127
BSC or SS	0	1	255
Token-Ring, frame-relay, or ISDN logical lines	0	0	127

[1] If t is not equal to 0 and m is less than or equal to 64, the default is 1; otherwise, the default is 0.

ml

Use: Token-Ring logical lines

Default: 6
Minimum: 0
Maximum: 128

For Token-Ring logical lines, ml specifies the maximum number of retries on a Token-Ring logical line established over a local Token Ring.

SDLC Links: On an SDLC link, an error condition exists when NCP does not receive positive indication that a frame sent was received successfully. When this occurs, NCP retransmits the frame.

If NCP is functioning as a secondary device on a line, t and n are ignored. The maximum number of retries is 128. You should specify 128 for m and 1 for n.

You can specify t and n individually for each station connected to the SDLC link represented by this LINE definition statement. Specify these values in RETRIES on the PU definition statement.

You can also specify t and n on a PU definition statement associated with a LINE definition statement for a backup SDLC local-channel-attached or local-remote link (that is, a PU definition statement from which you omit SUBAREA).

For physical units added by dynamic reconfiguration, t and n are ignored.

Retries of the following kinds of error conditions are determined by fixed parameters within NCP, not by the values you specify for RETRIES: information frames received out of sequence, information frames with data check errors, and unexpected responses (such as a request for initialization).

For casual connection, code m if you specify MODE=SECONDARY on the GROUP definition statement. t and n are not valid.

If you do not code the RETRIES keyword, the default is RETRIES=(15,0,0)

To prevent NCP from losing contact with an adjacent SDLC-link-attached controller while the adjacent controller is accessing its disk during a load or dump over the link, do the following:

1. Specify REPLYTO and RETRIES on the GROUP, LINE, or PU definition statement that defines the attachment to the adjacent controller so that the retry time is at least 3 minutes.

2. Set the link activity timer on the MOSS LKP function Common Options panel of the adjacent controller to a value equal to or greater than both of the following values:

   • REPLYTO on the GROUP definition statement that defines the attachment to the adjacent controller

   • The t suboperand of RETRIES on the GROUP, LINE, or PU definition statement that defines the attachment to the adjacent controller.

BSC or Start-Stop Lines: When NCP detects an error on a BSC or start-stop line, it retransmits the block that produced the error. When you specify

RETRIES=255, NCP continues retransmitting the block until it is received without error.

When a text-read error occurs, NCP does not repeat the retry sequence; that is, *t* and *n* do not apply to text-read errors.

Token-Ring Lines: RETRIES is used only for Token-Ring logical lines. *m* specifies the number of retries in each retry sequence for sessions established over a remote Token Ring. *ml* specifies the number of retries in each retry sequence for sessions established over a local Token Ring. If you do not code the RETRIES keyword, NDF adds RETRIES=(6,0,0,6) to the generation definition.

NTRI Frame Handler Logical Lines: RETRIES is used only for route determination processing.

Frame-Relay Logical Lines: *ml* is not valid. If you do not code the RETRIES keyword, NDF adds RETRIES=(6,0,0) to the generation definition.

If a substitute route for a PVC is defined in the frame-relay network, you may need to code the values of LOCALTO and RETRIES such that the combination of these two values is greater than the LMI error threshold of any node in the PVC. This gives the network time to switch to the substitute PVC.

Note: You can also code T1TIMER on the LINE definition statement to specify the LOCALTO value.

Frame-Relay Physical Lines: Only *m* is valid.

ISDN: RETRIES is used only for ISDN logical lines (when you specify ISDN=(LOGICAL,SUBAREA) or ISDN=(LOGICAL,PERIPHERAL)). *ml* is not valid. If you do not code the RETRIES keyword, NDF adds RETRIES=(6,0,0) to the generation definition.

RING
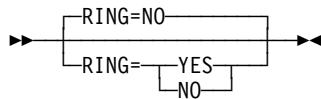
Use: Switched lines (NCP or EP)
Default: NO

Specifies whether the ring-indicator mode of an automatic answer operation is used. This decision depends upon the type of modem (data set) that connects the link to the controller. Determine from the modem supplier or installer whether the modem has a ring-indicator interface lead.

If the modem has the ring-indicator interface, code RING=YES. If it does not, code RING=NO (or omit the keyword). You must specify RING=NO if X21SW=YES on the GROUP definition statement.

Note: If you code LPDA2DL=YES on LINE, specify RING=YES. If you code LPDA2DL=YES and either CALL=IN or CALL=INOUT, RING=YES is the default and RING=NO is not valid. Note that CALL is a VTAM-only keyword when LNCTL=SDLC and the default is CALL=NO.

SDLCST

```
►►──┬────────────────────────────────────┬──►◄
    └─SDLCST=(──┬────────┬──┬──────────┬──)─┘
               └─plcst──┘  └─,slcst──┘
```
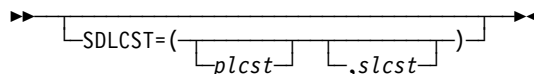
Use: Point-to-point subarea lines
Default: None
Examples: SDLCST=(N0ST001,N1ST004)
            SDLCST=(,N1ST004)

Specifies which SDLC selection table entry NCP is to use when it is operating in a
primary state (*plcst*) and which entry it is to use when it is operating in a secondary
state (*slcst*). (Parameters specified on the GROUP, LINE, and PU definition state-
ments associated with the link are used until NCP operates as a primary or sec-
ondary NCP.) The SDLCST keyword is required for links attached to a type 4
physical unit and is not valid for a type 1 or type 2 physical unit.

plcst and *slcst* are the names of SDLCST definition statements. See the
description of the SDLCST definition statement for more information on using an
SDLC selection table.

If you want to predefine the state of an NCP connected by this line, define one end
of the line as primary by coding SDLCST=*plcst*, and define the other end of the line
as secondary by coding SDLCST=(,*slcst*).

Token Ring, ISDN, and Frame Relay: Code the SDLCST keyword if you specify
one of the following conditions on the GROUP definition statement:

- ECLTYPE=(LOGICAL,SUBAREA)
- FRELAY=LOGICAL
- FRELAY=(LOGICAL,SUBAREA)
- ISDN=(LOGICAL,SUBAREA)

SECURE
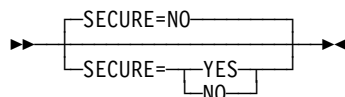
```
      ┌─SECURE=NO───────┐
►►──┬─┴─────────────────┴─┬──►◄
    └─SECURE=──┬─YES─┬─────┘
              └─NO──┘
```

Use: Switched, duplex, start-stop lines in network control mode
Default: NO

Specifies whether NCP uses the secure option for the start-stop line represented by
this LINE definition statement. Use of this option prevents access to an application
program's data by a start-stop terminal that dials the communication controller over
the line at the moment the existing connection to a different terminal is lost. If you
code SECURE=YES, NCP continuously monitors the data-carrier-detect signal from
the modem to give positive assurance that the switched connection still is estab-
lished.

SECURE is valid only if the LINE (or GROUP) definition statement specifies
DUPLEX=FULL and the GROUP definition statement specifies LNCTL=SS and
DIAL=YES.

SERVLIM

```
►►──┬─────────────────────────────────────────┬──►◄
     └─SERVLIM=(──┬──────────┬──┬───────────┬──)─┘
                   └─entries──┘  └─,seconds──┘
```
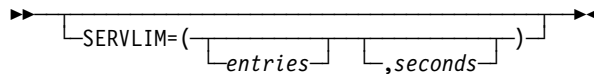
Determines how often NCP interrupts its regular scans of the service order table (SOT).

entries

> Use: Nonswitched multipoint lines and SDLC links in network control mode
>
> Default: See text
>
> Minimum: 1
>
> Maximum: See text

> **Start-Stop or BSC Lines:** Specifies the maximum number of entries in the SOT that NCP is to check for this line each time the program performs service seeking. The maximum is 254.
>
> If you omit SERVLIM and this LINE definition statement represents a start-stop or BSC line (LNCTL=SS, LNCTL=BSC), the program checks half of the table entries each time it performs service seeking. SERVLIM is valid only if you specify POLLED=YES on this definition statement. For BSC IBM 3270 lines, specify the highest number of devices attached to a cluster controller on the line.

> **SDLC Links:** Specifies the maximum number of regular scans of the SOT that NCP is to make for normal servicing of physical and logical units on the link before it makes a special scan of the table. The maximum is 255.
>
> Base your selection of a value for SERVLIM on:
>
> - The relative number of status commands the access method issues for the link served by the SOT
>
> - The relative importance of the following alternatives:
>
>   – A low value in SERVLIM gives NCP more frequent opportunities to fulfill accumulated status commands than does a higher value. Such status commands are fulfilled more promptly, but at the cost of frequent interruptions to normal servicing.
>
>   – A higher value in SERVLIM causes fewer interruptions to normal servicing of devices, but delays fulfillment of the status commands.
>
> - Experience.
>
> If you omit SERVLIM and this LINE definition statement represents an SDLC link (LNCTL=SDLC), a value of 4 (four regular scans of the table) is assumed.
>
> SERVLIM is not valid if you specify MODE=SECONDARY on the GROUP definition statement.

> **Attention:** You can minimize the possibility of throughput degradation by coding a high value in SERVLIM or a low value in REPLYTO, or both. Also, you can prevent the risk by arranging network operating procedures to avoid sending commands to physical units whose power is off.

LINE

Note: NCP performs a time out for any status command issued for the physical unit of an SDLC station whose power is off. The duration of this time out interval is the value of the REPLYTO keyword on the GROUP definition statement.

Normal servicing of physical units is interrupted during the time out interval because regular scanning of the SOT is suspended. Only when the specified interval expires does regular scanning (and therefore normal servicing) resume. A low value for SERVLIM (which causes relatively frequent special scans) and a high value for REPLYTO (resulting in long time out delays) results in serious degradation of message throughput on the SDLC link if status commands are received for SDLC stations whose power is off. (The time out interval recurs for each successive special scan.)

seconds

Use: SDLC links in network control mode
Default: See text
Minimum: 1.0
Maximum: 600.0

SDLC Links: Specifies the minimum time between special scans of the service order table.

If you do not specify a value for *seconds*, the frequency of special scans will be determined by *entries*. If you specify both *seconds* and *entries*, *seconds* determines the frequency of special scans.

SERVPRI

```
                ┌─SERVPRI=OLD──────┐
▶▶──────────────┼──────────────────┼──▶◀
                └─SERVPRI=─┬─OLD─┬──┘
                          └─NEW─┘
```
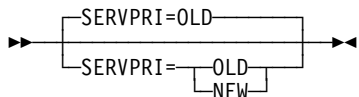
Use: Multipoint start-stop and BSC lines in network control mode
Default: OLD

Specifies whether NCP is to give priority to servicing current sessions (SERVPRI=OLD) or to establishing new sessions (SERVPRI=NEW) on the line.

SERVPRI is valid only for a nonswitched line in network control mode with multipoint line control.

SESSION

```
                ┌─SESSION=1────────┐
▶▶──────────────┼──────────────────┼──▶◀
                └─SESSION=sessions─┘
```
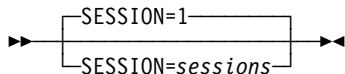
Use: Nonswitched, multipoint start-stop and BSC lines in network control mode
Default: 1
Minimum: 1
Maximum: 255

Specifies the number of sessions NCP will attempt to maintain concurrently on a nonswitched line in network control mode. The maximum number of sessions must not exceed the number of devices connected to the line.

SESSION does not limit the number of sessions if clustered stations are attached to the line and general polling is used. NCP cannot control the number of terminals that can respond to a general poll, and a separate session is established with each responding terminal.

If this line has IBM 3270 terminals attached, the value of *count* must be the sum of the number of cluster controllers and the number of terminals on the line. This value is also the default.

SPDSEL
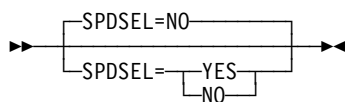
```
                  ┌─SPDSEL=NO──────┐
►►────────────────┴────────────────┴──►◄
              └─SPDSEL─┬─YES─┬─┘
                       └─NO──┘
```

Use: Start-stop and BSC lines in network control mode
Default: NO

Specifies whether the data rate of a dual-rate modem can be changed by request from the access method.

Code SPDSEL=YES to allow NCP to change the data rate of the modem. SPDSEL=YES is valid only if external clocking is used on the line. The data rate specified on SPEED must be the higher of the two data rates.

Code SPDSEL=NO or omit the keyword if the modem has one data rate, or if it has two data rates but NCP does not change the rate.

SPEED

```
►►──┬──────────────────────┬──►◄
    └─SPEED─┬─AUTO─┬─┘
            ├─RPQ──┤
            └─rate─┘
```
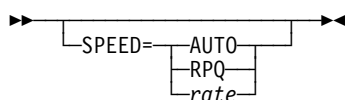
Use: NCP or EP
Default: None

Specifies the data rate for a line. For more information about the CLOCKNG keyword, see CLOCKNG on the LINE definition statement on page 2-208.

The SPEED keyword is required under any of the following conditions:

- When you code CLOCKNG=INT
- When you code CLOCKNG=DIRECT or CLOCKNG=(EXT,LIC)
- When you are using NPM to collect NCP performance data
- For X.21 lines
- For 3746 Model 900 SDLC lines
- For 3746 Model 900 frame-relay lines
- For SDLC peripheral and frame-relay physical peripheral lines that support APPN HPR data

AUTO

Specifies that NCP is to determine the speed for a telecommunication link for TWX autospeed lines. Specify SPEED=AUTO if you code DIAL=YES, LNCTL=SS, and TERM=TWX for this telecommunication link. Type an upper-case *X* as the first character entered after establishing a connection; you have 23.5 seconds to type an *X* before NCP disconnects the line.

EP: Enter a carriage return or type an uppercase *X* as the first character entered after establishing a connection.

RPQ

Specifies that a special speed has been selected and installed in the controller by Request Price Quotation (RPQ).

rate

Specifies the data rate for this line in bits per second. This data rate is the rate at which the devices communicate with the communication controller.

If you use NPM, specify SPEED to ensure that line utilization is correctly calculated. Define SPEED as the clocking rate provided by the external source. If this line is attached to a dual-rate modem, specify the higher of the two rates for SPEED.

If this LINE definition statement represents a multiple-terminal-access line used to communicate with terminals of different speeds, specify the speed of any of the terminals. For example, if terminals operating at speeds of 110 bps and 134.5 bps use this line, specify either SPEED=110 or SPEED=134.

If you code CLOCKNG=(EXT,LIC) on the LINE definition statement, code SPEED to specify the data rate. Code SPEED=9600 if the data rate is not listed in Table 2-76 and is less than 9600 bps. Code SPEED=55855 if the data rate is not listed in Table 2-76 and is greater than 9600 bps.

Table 2-76 (Page 1 of 2). Values for SPEED on the LINE Definition Statement

Data Rate (bps)	CLOCKNG=INT LNCTL=SS	CLOCKNG=INT LNCTL=BSC LNCTL=SDLC	CLOCKNG=DIRECT LNCTL=SS	CLOCKNG=DIRECT LNCTL=BSC LNCTL=SDLC
50	▪	▪	▪	▪
75	▪		▪	▪
100	▪		▪	▪
110	▪	▪	▪	▪
134	▪	▪	▪	▪
200	▪	▪	▪	▪
300	▪	▪	▪	▪
600	▪	▪	▪	▪
1200	▪	▪	▪	▪
2400	▪	▪	▪	▪
4800	▪	▪	▪	▪
9600	▪		▪	▪
19200	▪		▪	▪

[1] This speed is not valid for EP lines.
[2] This speed is valid only for HPTSS lines.

Table 2-76 (Page 2 of 2). Values for SPEED on the LINE Definition Statement

Data Rate (bps)	CLOCKNG=INT LNCTL=SS	CLOCKNG=INT LNCTL=BSC LNCTL=SDLC	CLOCKNG=DIRECT LNCTL=SS	CLOCKNG=DIRECT LNCTL=BSC LNCTL=SDLC
38400				▪
55855				▪
245760 [1]				▪
1474560 [1,2]				▪
1843200 [1,2]				▪
1200X/75R	▪			
75X/1200R	▪			

[1] This speed is not valid for EP lines.

[2] This speed is valid only for HPTSS lines.

Note: If you code CLOCKNG=EXT for a line that is not an X.21 network line, code a numeric *rate*; NDF will flag all other values as not valid.

Token Ring: If you code SPEED for Token-Ring physical resources, code SPEED=9600. If you do not code SPEED, NDF adds SPEED=9600 to the generation definition.

ESCON Links: The SPEED keyword does not determine the data rate for ESCON links. However, you can specify SPEED for ESCON physical and logical links to enable the network performance analyzer to estimate ESCON link utilization. It is recommended that you specify SPEED=144000000 initially and adjust this value as necessary.

APPN HPR: NCP uses the value of the SPEED keyword to determine the rate of flow for APPN HPR data on SDLC peripheral lines and on frame-relay physical peripheral lines. Code the SPEED keyword for SDLC peripheral and frame-relay physical peripheral lines that support APPN HPR data to ensure that the APPN HPR flow control mechanism works correctly.

Starting with VTAM V4R4 and NCP V7R6, the CAPACITY value from VTAM's topology database is used to determine link capacity and transmission time instead of the SPEED keyword. The value coded on the SPEED keyword is still used for earlier versions of VTAM and NCP.

ISDN Physical Lines: If you code SPEED for ISDN physical lines, code SPEED=2048000. If you do not code SPEED, NDF adds SPEED=2048000 to the generation definition.

ISDN B-Channels: If you code SPEED for ISDN B-channels, code SPEED=64000. If you do not code SPEED, NDF adds SPEED=64000 to the generation definition.

Frame-Relay and ISDN Resources: CLOCKNG=INT is not supported.

3746 Model 900:

The SPEED keyword is required for SDLC lines and for frame-relay lines.

| For lines with CLOCKNG=EXT coded, you should code a value that
| matches the actual line speed at which the line is running or is to run.
| This is required to insure the proper allocation of resources in the 3746
| Model 900 CLP. Just code the actual speed provided by the attached
| DCE. The minimum value is 600 bps; the maximum is 2048000 bps.

| For lines with CLOCKNG=INT or CLOCKNG=DIRECT, see Table 2-77
| for the SPEED values that you can code.

*Table 2-77. Values for SPEED on the LINE Definition
Statement for 3746 Model 900 SDLC and 3746 Model 900
Frame-Relay Lines*

Data Rate (bps)	CLOCKNG=INT	CLOCKNG=DIRECT
600	▪	▪
1200	▪	▪
2400	▪	▪
4800	▪	▪
9600		▪
19200		▪
32000		
38400		▪
55855		▪
64000		▪
256000		▪
512000		▪
1024000		▪
1536000		▪
2048000		▪

SPSHIFT

```
                  ┌─SPSHIFT=NO─────┐
►►─┬────────────────────────┬─►◄
   └─SPSHIFT=─┬─YES─┐
              └─NO──┘
```
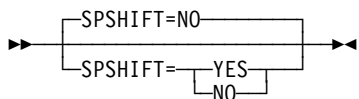
Use: Teletypewriter lines in network control mode
Default: NO

Specifies whether NCP interprets space characters received from 83B3, WU 115A,
or WTTY terminals as downshift characters when the line is in network control
mode. If SPSHIFT=YES, each space character received from a terminal causes
NCP to send all subsequent text characters to the host processor in their down-
shifted form.

If you code SPSHIFT=NO (or omit the keyword), NCP does not convert the text
characters, but instead sends them as they are received from the terminal.

SRETRY

```
               ┌─SRETRY=15───┐
►►─────────────┼─────────────┼──────►◄
               └─SRETRY=retries─┘
```
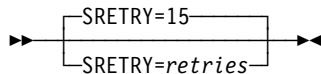
Default: 15
Minimum: 0
Maximum: 128

Specifies the number of retries to be attempted when a physical unit becomes secondary. SRETRY is valid only when DIAL=YES and LNCTL=SDLC on the GROUP definition statement. You can specify both SRETRY and RETRIES on a switched line.

SWMMTID

```
►►──────────────────────►◄
      └─SWMMTID=name─┘
```
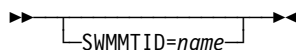
Use: Switched subarea lines
Default: None

Specifies the name of a switched SMMF table defined by a group of SWMMT definition statements. *name* must appear on the first SWMMT definition statement of a group of SWMMT definition statements that define a switched SMMF table.

For a switched subarea line, SWMMTID is required if you code MONLINK=YES or code MONLINK=CONTINUOUS on the LINE definition statement.

TADDR

```
►►──────────────────►◄
      └─TADDR=chars─┘
```
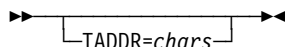
| Use: NCP or EP
| Default: See text

Specifies a symbolic link station address for this NCP when it is a type 4 physical unit on a subarea or peripheral link.

chars is the 2-digit hexadecimal representation of the EBCDIC character you code as 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 X'03' (the hexadecimal representation of the subarea address).

If you omit TADDR and the subarea address is 255, the default is 254.

If NCP is for a tributary controller on a BSC line, code TADDR=*chars*, where *chars* is the 2-digit hexadecimal representation of a single character (in transmission code). The character specified must be the same as the polling character specified in NCP for the controller that controls this line. If, for example, *A* is the polling character to which this tributary controller responds, you specify *A* as the tributary address in this LINE definition statement. In the other NCP, specify *A* in POLL on the appropriate TERMINAL definition statement. (See also ADDR on TERMINAL.)

Although the same character, *A*, is required in TADDR and POLL in the respective programs, the way you specify the character can differ. For POLL, code the hexadecimal representation of the EBCDIC letter *A* (POLL=C1). For TADDR on this LINE definition statement, code the hexadecimal representation of the transmission code bit pattern. If the transmission code used on the line is EBCDIC, the hexadecimal representation is the same (POLL=C1, TADDR=C1). However, if the transmission code is ASCII, code the hexadecimal representation of the ASCII letter *A* in TADDR (TADDR=41).

The polling character you assign to the tributary controller must have a bit pattern in which the third bit from the left is 0. Any uppercase alphabetical character in either EBCDIC or ASCII (except EBCDIC letters S–Z) meets this requirement. Any other bit pattern that meets this condition can be assigned also, whether or not that bit pattern represents a character. For example, all of the choices shown in Table 2-78 are valid polling characters for a tributary controller.

Table 2-78. Examples of Valid Polling Characters for a Tributary Controller

Code	Character	Binary Bit Pattern	Hex Bit Pattern
EBCDIC	A	1100 0001	C1
	!	0101 1010	5A
	(none)	0101 0100	54
ASCII	A	0101 0001	41
	[	0101 1011	5B

This keyword is required if you specify MODE=SECONDARY and do not code SDLCST on LINE for a secondary multipoint subarea link. TADDR is not valid if you code MODE=PRIMARY and do not code SDLCST on LINE for primary multipoint subarea links.

For peripheral links, TADDR is not valid if you code MODE=PRIMARY.

For casual connection, specify TADDR if you specify MODE=SECONDARY on the GROUP definition statement. TADDR is valid only if you specify LNCTL=SDLC on the GROUP definition statement. If you do not specify TADDR for a switched physical unit, NDF assigns a value between 1 and 126 to TADDR. NDF increments the default value by 1 each time NDF defaults TADDR.

The default for TADDR is the rightmost two digits of the hexadecimal representation of the subarea address coded on the SUBAREA keyword on the BUILD definition statement. This rule has two exceptions. If the rightmost two digits are X'00', the default is X'01'. If the rightmost two digits are X'FF', the default is X'FE'. The examples in Table 2-79 illustrate the calculation of the default.

Table 2-79. Examples of TADDR Defaults on the LINE Definition Statement

SUBAREA on BUILD	Hexadecimal Equivalent	Rightmost Two Digits	TADDR Default
1	0001	01	01
12346	303A	3A	3A
4608	1200	00	01
4607	11FF	FF	FE

TAILING

```
            ┌─TAILING=NO──────────┐
►►──────────┼─────────────────────┼──►◄
            └─TAILING=─┬─YES─┬─────┘
                       └─NO──┘
```
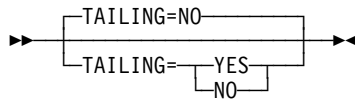
Default: NO

Specifies whether the line you are defining is attached to a second-level LPDA circuit.

Note the following coding considerations for TAILING:

- Code TAILING=YES if you define LPDATS=LPDA1, CHANLA=NO, and CHNLZ=YES. If you define LPDATS=LPDA2 you may code TAILING=YES.

- You can code TAILING=YES when MODE=SECONDARY on the GROUP definition statement.

- For casual connection, code TAILING=NO when MODE=SECONDARY on the GROUP definition statement.

- TAILING is not valid if you specify MODE=SECONDARY on the GROUP definition statement and TADDR on the LINE definition statement, and do not code SDLCST on the LINE definition statement for secondary multipoint subarea links.

TERM
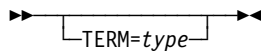
```
►►──┬──────────────┬──►◄
    └─TERM=type────┘
```

Use: NCP or EP
Default: None

Specifies the type of device NCP or EP will communicate with over this line. The device must be one of the types listed in Table 2-80 on page 2-283.

TERM is required if the line operates in emulation mode and you do not specify CUTYPE. If the line operates only in network control mode, specify TERM on the LINE or the TERMINAL definition statements. If you code CUTYPE on the LINE (or GROUP) definition statement, TERM is ignored. If different types of BSC devices (for example, IBM 3275 and IBM 3277) are attached to the line, specify only one of the device types.

Table 2-80 on page 2-283 shows that some types of devices are restricted to operation in network control mode. Devices of these types (System/7 or MTA terminals) must not be specified in TERM if the line operates in both network control and emulation mode.

Table 2-80 (Page 1 of 2). Values for TERM on the LINE Definition Statement

If Type of Device Is	Code TERM=
IBM 1030 Data Collection System	1030
IBM 1050 Data Communication System	1050
IBM 1060 Data Communication System	1060
IBM 1130 Computing System	1130
IBM 1800 Data Acquisition and Control System	1800
IBM System/360 System Model 20	2020
IBM System/360 System Model 25	2025
IBM 2260 Display Station	2260
IBM 2265 Display Station	2265
IBM 2701 Data Adapter Unit	2701
IBM 2715 Communications Terminal Model 1	2715
IBM 2740 Transmission Control Unit Model 2	2740-1
IBM 2740 Communications Terminal Model 2	2740-2
IBM 2741 Communications Terminal	2741
IBM 2770 Data Communications System	2770
IBM 2780 Data Transmission Terminal	2780
One of the following IBM 3270 Information Display Systems	
IBM 3275 Display Station[1]	3275
IBM 3276 Control Unit Display Station	3277
IBM 3277 Display Station[1]	3277
IBM 3278 Display Station	3277
IBM 3284 Printer[1]	3284
IBM 3286 Printer[1]	3286
IBM 3287 Printer	3284
IBM 3288 Line Printer	3286
IBM 3289 Line Printer	3286
IBM 3650 Retail Store System (in BSC mode)	SYS3
IBM 3660 Supermarket System (in BSC mode)	SYS3
IBM 3704 Communications Controller	3704
IBM 3735 Programmable Buffered Terminal	3735
One of the following IBM 3740 Data Entry Systems	
IBM 3741 Data Station	3741
IBM 3747 Data Converter	3747

[1] Not supported on switched lines. Note the following:
- Specify TERM=SYS3 on the TERMINAL, LINE, or GROUP definition statement if the line operates only in network control mode.
- Specify TERM=SYS3 on the LINE or GROUP definition statement if the line operates in both network control and emulation modes (alternately).
- Specify TERM=SYS7 on the LINE or GROUP definition statement if the line operates only in emulation mode.

Table 2-80 (Page 2 of 2). Values for TERM on the LINE Definition Statement

If Type of Device Is	Code TERM=
IBM 3767 Communications Terminal (in start-stop mode) supported as one of the following:	
2740 Model 1	2740-1
2740 Model 2	2740-2
2741	2741
IBM 3770 Data Communications System (in BSC mode)	2770
IBM 3780 Data Communications Terminal	3780
IBM System/370 System Model 125	3125
IBM System/370 System Model 135	3135
IBM System/3	SYS3
IBM System/7 (BSC version)[1]	—[1]
IBM System/7 (SS version)	2740-1
IBM System/32 (BSC version)	SYS3
IBM Communicating Magnetic Card Selectric Typewriter	2741
Western Union Teletypewriter Exchange Service IBM 3720 Communication Controller	TWX
World Trade Teletypewriter Terminals[1]	WTTY
Multiple Terminal Access (IBM 1050,2740,2741;TWX)[1]	MTA

[1] Not supported on switched lines. Note the following:
- Specify TERM=SYS3 on the TERMINAL, LINE, or GROUP definition statement if the line operates only in network control mode.
- Specify TERM=SYS3 on the LINE or GROUP definition statement if the line operates in both network control and emulation modes (alternately).
- Specify TERM=SYS7 on the LINE or GROUP definition statement if the line operates only in emulation mode.

TIMEOUT

```
►►──┬─────────────────────────┬──►◄
    └─TIMEOUT=──┬─seconds─┬────┘
                └─NONE────┘
```

Use: Channel links
Default: See text
Minimum: 10.0
Maximum: 840.0

Specifies how long NCP or EP is to wait for a response to an attention signal it sent to the host before initiating channel discontact (TIMEOUT=*seconds*), or that it is to wait an unlimited time for a response from the host (TIMEOUT=NONE).

Specify *seconds* either as an integral number of seconds or to the nearest 10th of a second. If you code DOS, VSE, OS, or MVS for TYPSYS on the BUILD definition statement, the default for TIMEOUT is 180.0 seconds. If you code TYPSYS=VM, the default for TIMEOUT is 420.0 seconds.

Attention: Specifying TIMEOUT=NONE results in non-responding terminals, long-term slowdown of the communication controller, and blocked virtual routes through the network if contact with the host is lost. You should not specify TIMEOUT=NONE if you have a complex network.

ESCON Links: The TIMEOUT keyword is valid only for logical lines. Do not code TIMEOUT for physical lines.

EP Standalone: TIMEOUT is valid only when NCPCA=ACTIVE is specified on the LINE statement.

TITIMER

```
               ┌─TITIMER=60──────────────┐
►►─────────────┼─────────────────────────┼───►◄
               └─TITIMER=time_interval───┘
```
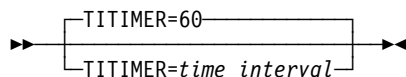
Use   Token-Ring, frame-relay, and ISDN physical lines
Default: 60
Minimum: 10
Maximum: 1500

Specifies the inactivity timer (TI) interval, in seconds, that NCP is to wait for a transmitted frame from an adjacent station before sending a poll command. *time_interval* can be expressed as a whole number, or to the nearest 10th of a second.

Note: If the TI timer value is too long, NCP can be delayed in detecting a failure in the adjacent station. If the TI timer value is too short, congestion can occur. In most cases, the default value is the best choice.

TRANSBUF

```
►►──────────────────────────────────────────►◄
        │                    ┌─D─┐     │
        └─TRANSBUF=(nn,──────┼───┼──)──┘
                             ├─D─┤
                             └─H─┘
```
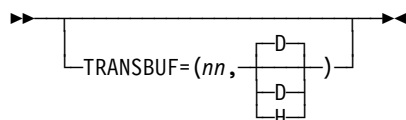
Use   3745 frame-relay physical lines
Default: 1,D for TSS lines; 20,D for HPTSS lines
Minimum: 1
Maximum: 4 for TSS lines; 20,D for HPTSS lines

Specifies the maximum number of frames that will be queued for a TSS or HPTSS adapter awaiting transmission over a 3745 frame-relay physical line.

TRANSBUF definition formats and definition rules are as follows:

- TRANSBUF=(*nn*,D) — D means that TRANSBUF is specified in decimal.
- TRANSBUF=(*nn*,H) — H means that TRANSBUF is specified in hexadecimal.

Decimal is assumed if you do not specify a value.

HPTSS Adapters: If the microcode does not support subarea and peripheral connections on a single line, the defined keyword value is ignored by NCP. Instead, NCP queues only one frame for the adapter.

TRANSFR
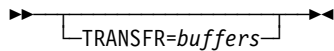
```
►►──┬─────────────────────┬──►◄
    └─TRANSFR=buffers─┘
```

Default: Value of TRANSFR on the BUILD definition statement
Minimum: 1
Maximum: 254

Specifies the number of NCP buffers corresponding to the maximum amount of data (the data transfer limit) that NCP can receive from the line during a single data transfer operation. If you code LNCTL=CA on the GROUP definition statement, TRANSFR specifies the maximum length PIU the channel adapter will accept.

Note: Be sure to spell TRANSFR without the letter *E*.

Attention: The value of TRANSFR must be the same for all lines in the transmission group; otherwise, link activation problems will occur.

If you do not inhibit subblocking, code a value of 2 or more for TRANSFR.

Be sure that the value you specify for TRANSFR lies between certain minimum and maximum values. The sections that follow tell you how to calculate the value of TRANSFR.

For channel links and SDLC links, the value of TRANSFR is automatically incremented by one.

Calculating Minimum Value of TRANSFR: To calculate the minimum value of TRANSFR for a BSC or start-stop line, do the following:

1. Determine the maximum amount of data to be received from the line represented by this LINE definition statement.

2. Add 54 bytes to the amount determined in Step 1.

3. Divide the sum by NCP buffer size (BFRS on the BUILD definition statement).

4. Round the result to the next higher integer.

To calculate the minimum value of TRANSFR for an SDLC line, do the following:

1. Determine the maximum amount of request/response unit data to be received from the line represented by this LINE definition statement.

2. Add 47 bytes to the amount determined in Step 1.

3. Divide the sum by NCP buffer size (BFRS on the BUILD definition statement).

4. Round the result to the next higher integer.

To calculate the minimum value of TRANSFR for a channel link (LNCTL=CA on the GROUP definition statement), do the following:

1. Determine the maximum amount of request/response unit data to be received from the line represented by this LINE definition statement.

2. Add 51 bytes to the amount determined in Step 1.

> 3. Divide the sum by NCP buffer size (BFRS on the BUILD definition statement).
>
> 4. Round the result to the next higher integer.

If this LINE definition statement defines a subarea link between two type 4 physical units, the value of TRANSFR times the NCP buffer size (BFRS on the BUILD definition statement) must equal or exceed 1050 bytes.

If this LINE definition statement defines a channel link (LNCTL=CA on the GROUP definition statement), the value of TRANSFR times the NCP buffer size must equal or exceed 1296 bytes.

If this LINE definition statement defines an SDLC link for a type 1 or type 2 physical unit, the value of TRANSFR times the NCP buffer size must equal or exceed 559 bytes.

If you use the VTAM MODIFY LOAD command to add or replace a load module on the communication controller hard disk, the value of TRANSFR times the NCP buffer size must equal or exceed 2048 bytes.

Calculating Maximum Value of TRANSFR: To calculate the maximum value of TRANSFR, do the following:

> 1. Determine which host processors in the network receive the data.
>
> 2. Calculate the data transfer limit, in bytes, imposed by the access method in each destination host processor. Multiply the maximum number of buffers the access method allocates for receiving data from NCP (MAXBFRU on the VTAM HOST definition statement) times the size of each buffer (UNITSZ on the VTAM HOST definition statement). From this value, subtract the buffer pad for the access method (BFRPAD on the VTAM HOST definition statement).
>
> 3. Determine which of the data transfer limits calculated this way is the smallest and divide that limit by the NCP buffer size (BFRS on the BUILD definition statement). The result (ignoring any fractional remainder) is the maximum value you should code for TRANSFR.

Type 7 Channel Adapter: Specify a value as close as possible to the size of the largest PIU NCP will receive. Specifying too large a value can degrade performance and waste NCP buffer storage.

Frame Relay: TRANSFR is not valid for frame relay.

Multilink Transmission Groups: The value of TRANSFR can affect the segment size of a MLTG that contains this line. For more information, refer to the discussion of MLTG segment size in the *NCP, SSP, and EP Resource Definition Guide*.

ESCON Links: The TRANSFR keyword is valid only for logical lines. TRANSFR is not valid for physical lines.

TRSPEED

```
►►──┬──────────────────────────┬──►◄
    └─TRSPEED=─┬──4──┬─┘
              └──16──┘
```
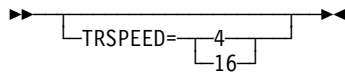
Use: Token-Ring physical lines
Default: See text

Specifies the speed of data on the Token Ring in megabits per second. Keep in mind that all adapters on the same Token Ring must use the same speed. See Table 2-81 for valid TRSPEED values.

Table 2-81. Values for TRSPEED on the LINE Definition Statement

Adapter	TRSPEED Values	Default
TIC1	4	4
TIC2 or TIC3	4, 16	16

3746 Model 900 Token Ring: TRSPEED=4 is not valid if ADDRESS=2080.

TYPE

```
►►──┬────────────────────┬──►◄
    └─TYPE=─┬──NCP──┬─┘
           ├──PEP──┤
           └──EP───┘
```
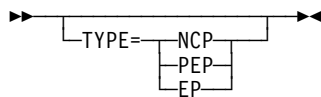
Default: See text

Specifies whether this line will operate in network control mode (TYPE=NCP), emulation mode (TYPE=EP), or both (TYPE=PEP).

Follow these rules for coding TYPE:

- If you code TYPE=PEP on the GROUP definition statement and this line will operate only in network control mode, code TYPE=NCP on this LINE definition statement.

- If you code TYPE=PEP on the GROUP definition statement and this line will operate only in emulation mode, code TYPE=EP on this LINE definition statement.

- If you code USE=SPARE or USE=REDEF for this line, only TYPE=NCP is valid. Let this LINE definition statement default to TYPE=NCP, or explicitly code TYPE=NCP on this LINE definition statement or on the GROUP definition statement for this line.

- In any other case, you do not need to code TYPE on this LINE definition statement.

EP Standalone: TYPE=EP is the only valid value and is the default.

Follow these rules for coding other keywords on this LINE definition statement:

- If this line will operate only in network control mode, code only those keywords that are valid for network control mode.

- If this line will operate only in emulation mode, code only those keywords that are valid for emulation mode. Keywords that apply only to network control mode are ignored.

- If this line will operate in both network control mode and emulation mode, you can code any keyword.

Table 2-82 shows the effect on TYPE when you code certain keywords on the GROUP definition statement.

Table 2-82. NDF Values for TYPE on the LINE Definition Statement

When You Code This on GROUP	The Effect on TYPE Is
ECLTYPE	NDF adds TYPE=NCP. The only valid value is TYPE=NCP.
ETHERNET=PHYSICAL	The default is TYPE=NCP.

T1TIMER
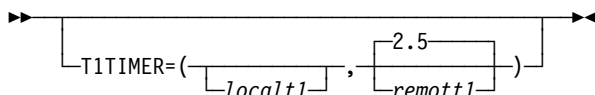
Use: Token-Ring logical subarea resources
       Frame-relay logical resources
       ISDN logical resources

Specifies, in seconds, the reply timer (T1) values for stations connected to ISDNs, Token-Ring networks, or frame-relay networks.

For Token-Ring and frame-relay resources, you can also specify these values using LOCALTO and REMOTTO. See LOCALTO on the LINE definition statement on page 2-239 and REMOTTO on the LINE definition statement on page 2-268 for specific information.

localt1

Default: See text
Minimum: 0.6
Maximum: 20.0

Specifies the reply timer value for stations connected to ISDNs, Token-Ring networks, or frame-relay networks.

Token Ring: *localt1* specifies the Token-Ring reply timer (T1) value for stations connected to a local Token Ring. *localt1* is valid only for logical subarea lines. (Code *localt1* on T1TIMER on the BUILD definition statement to specify a T1 value for logical peripheral lines.)

The value of *localt1* must not be greater than the value of *remott1*. The default value for *localt1* is either the value of *remott1* on this T1TIMER on the LINE definition statement or the value of *remott1* on T1TIMER on the BUILD definition statement, whichever is less.

Frame Relay: *localt1* specifies the frame-relay reply timer value for stations connected to a frame-relay network. If you do not code *localt1* on either the LINE or GROUP definition statement, the default is 6.0. The

value of T1TIMER on the BUILD definition statement is not used for frame relay.

Important: It is important to code a value for *localt1* that reflects the maximum amount of time that it takes to receive a solicited response from the adjacent link station. Factors that can affect this time include line speed, propagation delay, maximum frame size, delay caused by intermediate frame handlers, and the number of DLCIs supported by the associated physical line that could create queuing delays for the FRTEs. If the value you code for *localt1* is too small, the FRTEs can lose synchronization and fail the connection. This is indicated by an INOP 01 and a link event or an alert that indicates a FRMR condition. If this occurs, code a larger *localt1* value for this FRTE connection.

If a substitute route for a PVC is defined in the frame-relay network, you may need to code the values of *localt1* and RETRIES such that the combination of these two values is greater than the LMI error threshold of any node in the PVC. This gives the network time to switch to the substitute PVC.

An indication of INOP 02 can occur for a frame-relay logical line if it is used to perform a remote load. To avoid the condition where the INOP 02 occurs prior to the +RSP(IPFINAL), the LMI error threshold time interval for the link station subport should be at least 3 times the T1 timer interval for the logical subarea line. The LMI error threshold time interval is $t391 \times n392$, where $t391$ is a time interval defined by TIMERS on the PU definition statement for the link station subport, and $n392$ is an error count threshold defined by ERRORT on the same PU definition statement.

NTRI Frame Handler: *localt1* is *not* valid for NTRI frame handler logical lines. (Code *localt1* on T1TIMER on the BUILD definition statement to specify a *localt1* value for NTRI frame handler logical lines.)

ISDN: *localt1* specifies the ISDN reply timer value for stations connected to an ISDN. If you do not code *localt1* on either the LINE or GROUP definition statement, the default is 6.0. The value of T1TIMER on the BUILD definition statement is not used for ISDN.

Important: It is important to code a value for *localt1* that reflects the maximum amount of time that it takes to receive a solicited response from the adjacent link station. Factors that can affect this time include line speed, propagation delay, maximum frame size, and the number of DLCIs supported by the associated physical line that could create queuing delays for the adjacent link station. If the value you code for *localt1* is too small, the adjacent link stations can lose synchronization and fail the connection. This is indicated by an INOP 01 and a link event or an alert that indicates a FRMR condition. If this occurs, code a larger *localt1* value for this adjacent link station connection.

remott1

Default:	2.5
Minimum:	0.6
Maximum:	20.0

Specifies, in seconds, the reply timer (T1) value for stations connected to a remote Token Ring. You can specify this value as an integer or with one decimal place.

Token Ring: *remott1* is used for Token-Ring logical subarea resources only.

Frame Relay: *remott1* is *not* used for frame-relay logical resources.

ISDN: *remott1* is *not* used for ISDN logical resources.

T2TIMER

Use: Token-Ring, frame-relay, and ISDN logical lines

Specifies the acknowledgment timer (T2) values and the maximum number of I-frames that NCP can receive without sending an acknowledgment. This keyword is valid only if you code ECLTYPE=(LOGICAL,SUBAREA), FRELAY=LOGICAL, ISDN=(LOGICAL,SUBAREA), or ISDN=(LOGICAL,PERIPHERAL) on the GROUP definition statement.

If possible, consider the value of the reply timer (T1) used by the adjacent station when determining the value for the receiver acknowledgment timer. The acknowledgment timer should be about one-tenth as long as the reply timer for the adjacent station to ensure that the adjacent station will receive the delayed acknowledgment before its reply timer expires.

T2TIMER can be coded on the PU definition statement for frame-relay and ISDN logical lines and Token-Ring logical subarea lines. This tuning parameter **must** be coded on the PU definition statement for switched support via VTAM.

localt2

Default: See text
Minimum: 0.0
Maximum: 2.0

Token Ring: Specifies the timer value used for connections over a local Token Ring. Valid values are in intervals of a tenth of a second. The default is the value coded for *localt2* on the BUILD definition statement. A value of 0 indicates that the timer is not to be used for this line.

Frame-Relay Logical Lines: Specifies the timer value for connections through a frame-relay network. For logical subarea lines (FRELAY=(LOGICAL,SUBAREA) on the GROUP definition statement), the default is 1.0 if the line number of the corresponding physical line is in the range 0–895; the default is 0.2 if the line number of the corresponding physical line is in the range 1024–1039. For frame-relay logical peripheral lines, the default is 0.0.

The *localt2* value on the BUILD definition statement is not the default used for frame-relay lines.

A value of 0.0 indicates that the timer is not to be used for this line.

3746 Model 900 Frame-Relay Logical Lines: For logical subarea lines, if the speed of the corresponding physical line is greater than or equal to (≥) 256KB, the default value is 0.2. If the corresponding physical line speed is less than (<) 256KB, the default value is 1.0. For logical peripheral lines the default value is 0.0.

ISDN: The default value is 1.0 for ISDN logical subarea lines and 0.0 for ISDN logical peripheral lines.

remott2

Default: See text
Minimum: 0.0
Maximum: 2.0

Token Ring: Specifies the timer value used for connections over a remote Token Ring. Valid values are in intervals of a 10th of a second. The default value is the value coded for *remott2* on the BUILD definition statement. A value of 0 indicates that the timer is not to be used for this line.

Frame Relay: *remott2* is not valid for frame relay.

ISDN: *remott2* is not valid for ISDN.

n3

Default: See text
Minimum: 1
Maximum: 127

Token Ring: Specifies the maximum number of I-frames that can be received without sending an acknowledgment. The default is the value coded for *n3* on the T2TIMER keyword on the BUILD definition statement.

Frame-Relay and ISDN Logical Lines: Specifies the number of I-frames that can be received without sending an acknowledgment. If *n3* is not coded on either the LINE or the GROUP definition statement, the default is 2, not the value coded on the BUILD definition statement.

UACB

```
►►──┬─────────────────────────────┬──►◄
    └─UACB=(name1─┬──────────┬─)─┘
                  └─,name2─┘
```

Use: User-written line control, NTRI, and frame relay
Default: None

Specifies the names of the user, NTRI, or frame-relay adapter control blocks associated with this line. UACB is required when you code LEVEL2 or LEVEL3 on the GROUP definition statement. Otherwise, UACB is not valid.

name1 and *name2* are the names of the user or NTRI adapter control blocks for this line.

Do not code a value for UACB when you code AUTOCOPY on the LINE definition statement. NDF will supply a unique name for UACB during the generation process.

NTRI: Specify any valid symbol for UACB. If you do not code UACB, NDF adds UACB with an appropriate value to the generation definition.

Frame Relay: If you do not code UACB for a frame-relay logical line, NDF adds UACB=X$L*n*B to the generation definition. (*n* is a counter that is incremented for each frame-relay logical line.)

3746 Model 900 Token Ring: The UACB keyword is ignored for 3746 Model 900 Token-Ring resources.

3746 Model 900 Frame Relay: The UACB keyword is ignored for 3746 Model 900 frame-relay resources.

UNITXC

```
        ┌─UNITXC=YES─────┐
►►──────┤                ├──►◄
        └─UNITXC=─┬─YES─┬─┘
                  └─NO──┘
```
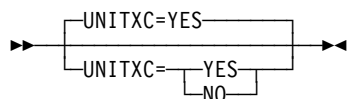
Use: Start-stop and BSC lines in emulation mode
Default: YES

Specifies whether NCP or EP signals unit-exception status to the host processor when the program receives an EOT from the line.

Usually, you can specify UNITXC=YES (or omit the keyword), which causes the control program to signal unit-exception status upon receiving an EOT. However, if read and write commands within the access method are command chained, specify UNITXC=NO. UNITXC=NO, by suppressing the unit-exception indication, prevents the command chain from being broken. (Unit-exception status always breaks the command chain.)

Note: Specify UNITXC=NO for IBM 2741 terminals equipped with the break feature if you require channel end and device end (CE and DE) rather than the normal ending sequence (CE, DE, UE).

USE

```
        ┌─USE=NCP────────┐
►►──────┤                ├──►◄
        └─USE=─┬─NCP───┬─┘
               ├─EP────┤
               ├─SPARE─┤
               └─REDEF─┘
```
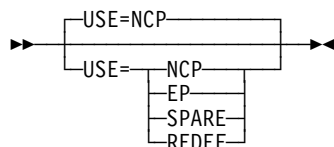
Use: For TYPE=PEP: Lines operable alternately in network control and emulation modes
            For TYPE=NCP: Lines that can alternately be spare lines or redefinable lines (or neither)
Default: NCP

When TYPE=PEP, specifies whether this line is to operate initially in network control mode (USE=NCP) or emulation mode (USE=EP). A request from the access method to NCP or EP switches a line from one mode to the other.

Note: The USE keyword is not valid for LNCTL=CA.

EP Standalone: The USE keyword does not apply to EPSA.

When TYPE=NCP, specifies whether this line is initially a spare line, a redefinable line, or neither.

- Code USE=SPARE to define a line as initially a spare line.

- Code USE=REDEF to define a line as initially a redefinable line.

- If you code USE=NCP, the line is not a redefinable or a spare line. It cannot be changed to a spare or redefinable line.

A spare line cannot be activated. It must be changed to a redefinable line to be activated. You change a spare line to a redefinable line, or a redefinable line to a spare line, by using NTuneMON with NTuneNCP.

A spare or redefinable line is assigned to a specific physical line and cannot be used by any other physical line. See the chapter about adding SDLC lines in the *NTuneMON V2R5 User's Guide* for a complete description of spare and redefinable lines.

USE=SPARE or USE=REDEF is valid only if:

- You code TYPE=NCP on this LINE definition statement.
- This line is a 3745-attached SDLC peripheral line.
- You include NTuneNCP in your NCP generation. If NTuneNCP is not included, you cannot change a spare line to a redefinable line, and, consequently, the line cannot be activated.

Do not code USE=SPARE or USE=REDEF on the GROUP definition statement. It is not valid.

Code only one spare or redefinable line per GROUP definition statement.

V25BIS

```
                 ┌─V25BIS=NO────────────────────┐
►►───────────────┼──────────────────────────────┼───►◄
                 └─V25BIS=──┬─(YES,─┬─DLSS───┬─)──┘
                            │       └─DLSDLC─┘    │
                            └─NO─────────────────┘
```
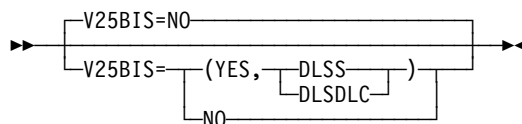
Use: V.25 bis
Default: NO

Indicates whether the modem for the NCP line or group of lines being defined supports V.25 bis protocol and identifies which line protocol is used for the dial connection (SDLC or start-stop).

Code V25BIS=(YES,DLSS) if you use V.25 bis and your modem supports start-stop line protocol during the dial phase. Code V25BIS=(YES,DLSDLC) if you use V.25 bis and your modem supports SDLC line protocol during the dial phase.

Note the following when you code V25BIS=(YES,DLSS) or V25BIS=(YES,DLSDLC):

- Code DIAL=YES.

- Code AUTO=YES to initiate proper VTAM commands. You must code AUTO=YES for V.25 bis call-out modem lines (CALL=OUT or CALL=INOUT).

• Code the REDIAL keyword for this line. If you do not code it, all suboperands default to 0.

If you code V25BIS=NO for a switched line, do not code the second suboperand.
Attention: If you code RING=YES and this line is attached to a modem that supports V.25 bis, NCP may detect a hardware error when a call is received from a device other than a modem. This will result in an INOP02 status for the line.

BSC: When you dial out to a BSC device using V.25 bis, you must code CLOCKNG=INT and SPEED=*modem speed* if your modem does not support asynchronous clocking (IBM 7855).

EP: V.25 bis switched line protocol is supported for emulation mode. You may code V25BIS on lines with TYPE=EP. The default is V25BIS=NO. When you specify V25BIS=(YES,DLSS) or V25BIS=(YES,DLSDLC), you must code DIAL=YES, LNCTL=SS, or LNCTL=BSC on the GROUP definition statement and AUTO=YES on this LINE definition statement.

YIELD

```
              ┌─YIELD=YES─┐
►►──┬───────────────────┬──►◄
    └─YIELD=──┬─YES─┬───┘
              └─NO──┘
```
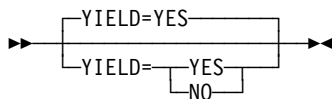
Use: BSC lines in network control mode
Default: YES

Specifies the controller as the secondary device on a nonswitched point-to-point (contention) line. YIELD applies only to line operation in network control mode.

Code YIELD=YES or do not code YIELD if the controller yields to the primary device when contention occurs. Code YIELD=NO if the controller is the primary device.

YIELD is valid only for a BSC point-to-point line (LNCTL=BSC is coded on the GROUP definition statement and POLLED=NO is coded or POLLED is not coded on the LINE or the GROUP definition statement).

LU Definition Statement

Usage: SDLC and channel-attached devices. One for each logical unit associated with a device on a nonswitched telecommunication link.

Ethernet-Type LANs: The LU definition statement is not valid for Ethernet-type LANs.

Internet Protocol: The LU definition statement is required for NCP connectionless SNA transport (NCST) definitions.

Frame Relay: The LU definition statement is not valid for frame relay.

ESCON Links: The LU definition statement is not valid for ESCON physical links.

3746 Model 900 Token Ring: The LU definition statement is not valid for 3746 Model 900 Token Ring.

Position: Must follow PU definition statement or other LU definition statement associated with same device.

Keywords: Table 2-83 shows the keywords described under this definition statement.

Table 2-83 (Page 1 of 2). Keywords on the LU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
INTFACE	2-299				
LOCADDR	2-300	Yes			
LUCB	2-301		G,L,PU		
LUDR	2-302		G,L,PU		
LUFVT	2-302		G,L,PU		
LUNTFY	2-303		G,L,PU	1	
MAXCOLL	2-303		G,L,PU		
MAXSESS	2-304		B	65535	
MAXTP	2-304		G,L,PU		
NETID	2-305				
NEXT	2-305				
NPACOLL	2-306				
NUMSESS	2-307		G,L,PU		
PACING	2-307		G,L,PU	(1,,ADAP)	
REMLU	2-308				
RESSCB	2-309			0	

Sift:

B   BUILD definition statement
G   GROUP definition statement
L   LINE definition statement
PU   PU definition statement

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-83 (Page 2 of 2). Keywords on the LU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
UCCB	2-309				
VPACING	2-309		G,L,PU	(0,,ADAP)	

Sift:

B   BUILD definition statement
G   GROUP definition statement
L   LINE definition statement
PU   PU definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements:
You can define some keywords of the LU definition statement on the PU, LINE, or GROUP definition statement instead of on the LU definition statement. Table 1-3 on page 1-12 shows which of the keywords you can code at a higher level.

Network Performance Analyzer Resources: If you specify NPARSC=YES on the GROUP definition statement, all keywords specified on this LU definition statement are ignored except for MAXCOLL, MAXTP, and NPACOLL. Up to eight LU definition statements may be coded, thus allowing multiple NCP-NPM sessions. One NCP-NPM session is the primary session and the others are secondary sessions. NCP uses the secondary sessions to transport session accounting data, request network address assignments (RNAAs), and FNAs, but not to transport performance data.

Channel Links: Do not code an LU definition statement if you code LNCTL=CA on the GROUP definition statement and define a channel subarea link (PUTYPE=5) on the PU definition statement. You must code an LU definition statement if you define a channel peripheral link (PUTYPE=2) on the PU definition statement.

If your generation definition contains predefined independent logical units, do one of the following:

- Delete the LU definition statements that predefine each independent logical unit and define them as cross-domain resources (CDRSCs) in the VTAMLST. Increase the sizes of dynamic reconfiguration pools for independent logical unit resources.

- Specify GENILU=NO on the BUILD definition statement so that the generation process does not generate resource control blocks and element addresses for independent logical units with LUDR=YES.

VTAM Keywords: The following diagrams show the VTAM keywords that can be coded on this definition statement. These keywords provide information to the VTAM initialization process. They are not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for descriptions of these keywords and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

VTAM keywords not listed below can be specified by coding VTM. as a prefix in front of the VTAM keyword.

```
►►──┬──────────────────────────────────────────────┬──►◄
     └─ASLENT=associated_LU_table_entry_name─┘
```

```
►►──┬──────────────────────────────────────────┬──►◄
     └─ASLTAB=associated_LU_table_name─┘
```

```
►►──┬────────────────────────────────────────────────┬──►◄
     └─DLOGMOD=default_logon_mode_entry_name─┘
```

```
     ┌─ENCR=NONE─────────────┐
►►──┼───────────────────────┼──┬───────┬──►◄
     └─ENCR⁽¹⁾=─┬─NONE─┐       
                ├─REQD─┤
                ├─SEL──┤
                └─OPT──┘
```

¹ Valid for MVS only

```
►►──┬──────────────────────────────┬──►◄
     │            ┌─,───────┐        │
     └─FEATUR2=(──▼─keyword─┴──)─┘
```

```
     ┌─ISTATUS=ACTIVE───────────┐
►►──┼──────────────────────────┼──►◄
     └─ISTATUS=─┬─ACTIVE───┐
                └─INACTIVE─┘
```

```
►►──┬──────────────────────────────────────────┬──►◄
     └─LOGAPPL=application_program_name─┘
```

```
►►──┬──────────────────────────────────┬──►◄
     └─LOGTAB=interpret_table_name─┘
```

```
►►──┬──────────────────────────────────────────┬──►◄
     └─MDLENT=model_name_table_entry_name─┘
```

```
►►──┬──────────────────────────────────────┬──►◄
     └─MDLTAB=model_name_table_name─┘
```

```
►►──┬──────────────────────────────────────┬──►◄
     └─MODETAB=logon_mode_table_name─┘
```

```
►►──┬─REGISTER=─┬─NO──────┐──┬──►◄
                ├─NETSRVR─┤
                └─DIRSRVR─┘
```

```
►►──┬──────────────────┬──►◄
     └─SPAN=span_name─┘
```

```
     ┌─SSCPFM=FSS─────────┐
►►──┼────────────────────┼──►◄
     └─SSCPFM=─┬─FSS────┐
              ├─USSSCS─┤
              ├─USS3270─┤
              ├─USS3275─┤
              ├─USS3780─┤
              └─USSNTO─┘
```

```
►►─────────────────────────►◄
        └─TERM=type─┘
```

```
►►─────────────────────────────────────────►◄
        └─USSTAB=USS_definition_table_name─┘
```

Keyword Descriptions

symbol

```
►►───symbol───►◄
```

Use: Required when you code the LU definition statement
Default: None

Provides a name for the logical unit. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates. (This name must not be specified for ORDER on the SERVICE definition statement.) If AUTOLINE is specified, you must embed the first suboperand of the ADDRESS keyword on the LINE definition statement in the symbol.

INTFACE

```
►►──────────────────────────────►◄
        └─INTFACE=(name──────)─┘
                      └─,mtu─┘
```
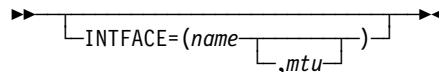
Use: Required for Internet Protocol

Specifies an internet router interface name for an NCST session. The interface name defined is referenced on the IPLOCAL, IPOWNER, and IPROUTE definition statements to correlate internet route table entries with an NCST logical unit. The first logical unit you define for a particular interface name is used as the primary NCST session and all others are used as backups.

Note: The INTFACE keyword is valid only when you code NCST=IP on the GROUP definition statement.

name

Default: None

Specifies the interface name, which must be no more than 8 characters in length. The name you specify must be different from the name specified for INTFACE on any Ethernet-type LINE definition statement. The name can, however, be the same as the name specified for INTFACE on other NCST LU definition statements.

mtu

Default: See text
Minimum: 68
Maximum: 65535

Specifies the *maximum transfer unit*, which is the maximum IP datagram size, in bytes, supported by this internet router interface. *mtu* should be coded only on the first LU definition statement for an NCST logical unit with a particular name.

For primary sessions, the default for *mtu* is 2048. For backup sessions, the default for *mtu* is the value coded for the primary session.

Specifying a small *mtu* for large IP datagrams can result in excessive fragmentation, which can degrade NCP performance.

For an LU-LU session where one logical unit resides at the host, the value of *mtu* plus 29 should be less than or equal to the maximum transfer size defined for the channel to the host (determined by MAXBFRU and UNITSZ on the VTAM HOST definition statement). Otherwise you may receive sense code X'800A'.

LOCADDR

►►──LOCADDR=*address*──►◄

Use: Required when you code the LU definition statement
Default: None

Specifies the local address (in decimal, without leading zeros) of the logical unit. The range of addresses for a logical unit associated with a type 1 physical unit is 0 to 63. The range of addresses for a logical unit associated with a type 2 physical unit is 1 to 255. The range of addresses for a logical unit associated with a type 2.1 physical unit is 0 to 255. (A type 2.1 physical unit has PUTYPE=2 and XID=YES coded on the PU definition statement.)

For LU definition statements with a nonzero value for LOCADDR, specify the addresses in ascending order: the LU definition statement specifying the lowest address must appear first (following the PU definition statement), and the LU definition statement specifying the highest address must appear last.

Code LOCADDR=0 for independent logical units. Independent logical units are valid only on type 2.1 physical units. The number of logical units you define with LOCADDR=0 is limited only by the controller storage size. You can code LU definition statements with LOCADDR=0 in any order following the PU definition statement.

Note: Some devices have a more restricted range of addresses than those listed. Consult the appropriate component description manual for these restrictions.

User-Written Channel Code: If this LU definition statement is in a line group that supports user-written channel handling code (CHANLNK=YES on the GROUP definition statement), LOCADDR specifies the subchannel address used by that user-written code.

address is the hexadecimal subchannel address and must be within the range specified by HICHAN and LOCHAN for the channel adapter position specified by ADDRESS on the LINE definition statement for this line.

LOCADDR=00 is not valid. See the information about LOCADDR in the description of UCHAN on the BUILD definition statement.

The subchannel address specified for this logical unit must not be the same as a subchannel address associated with:

- Any EP line on the channel adapter

- Any other logical unit on this line

- Any other program facility (for example, dynamic dump) using this channel adapter

Specify the subchannel addresses in ascending order; code the LU definition statement with the lowest subchannel address first (following the PU definition statement) and the LU definition statement with the highest address last.

Table 2-84 shows the effect on LOCADDR when you code certain keywords on the GROUP and LINE definition statements.

Table 2-84. NDF Values for LOCADDR on the LU Definition Statement

When You Code This	The Effect on LOCADDR Is
GROUP: ECLTYPE=PHYSICAL and LINE: ADAPTER=TIC1 or TIC2	NDF adds LOCADDR=0. The only valid value is LOCADDR=0.
GROUP: NCST=IP	NDF adds LOCADDR=1.[1]

[1] When NCST=IP, NDF adds incremented LOCADDR values to each successive LU definition statement that follows a PU definition statement. For example, NDF adds LOCADDR=1 to the first LU definition statement, LOCADDR=2 to the second LU definition statement, LOCADDR=3 to the third LU definition statement, and so on.

LUCB

```
►►──┬─────────────────────────┬──►◄
    │            ┌─,◄───────┐  │
    └─LUCB=(──▼─name─┴──)──┘
```
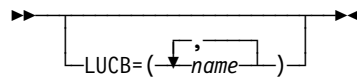
Use: User-written code and Internet Protocol
Default: None

Specifies the names of up to nine user-defined control blocks to be associated with this virtual logical unit. LUCB is valid only if you code VIRTUAL=YES on the GROUP definition statement.

If you specify NUMSESS, code two names: (1) the name of the NAU control block and (2) the name of the half-session control block. If you omit NUMSESS, you can specify one to nine names.

The names you specify for LUCB are related by position to the names you specify for the LUFVT keyword. For each name you specify for LUCB, you must define a corresponding name in the same relative position for LUFVT.

Internet Protocol: If you specify NCST=IP on the GROUP definition statement and you do not code LUCB on an associated LU definition statement, NDF adds LUCB with an appropriate value to the generation definition.

LUDR

```
►►──┬────────────────┬──►◄
    └─LUDR=─┬─NO──┬──┘
           └─YES─┘
```
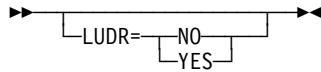
Use: Dynamic reconfiguration
Default: Value of PUDR on the PU definition statement

Specifies whether this logical unit can be deleted from the network using the dynamic reconfiguration function. LUDR=YES is valid only if dynamic reconfiguration is supported. LUDR=NO is not valid if PUDR=YES.

If LUDR=NO is coded or is used by default, NDF generates a resource control block and element address for this LU definition statement regardless of the value of GENILU on the BUILD definition statement.

NTRI: LUDR is not valid for NTRI logical resources. Only LUDR=NO is valid for physical resources and is the default.

APPN: You do not need to code LUDR=NO to ensure that a logical unit is available for a session with an APPN node. Instead it is recommended that you code the LUDRPOOL definition statement and ADDSESS, MAXSESS, and AUXADDR on the BUILD definition statement with values large enough for a dynamically added logical unit, and specify RESSCB when the logical unit is dynamically defined.

LUFVT

```
►►──┬──────────────────────┬──►◄
    │        ┌─,─────┐      │
    └─LUFVT=(─▼─name─┴─)────┘
```
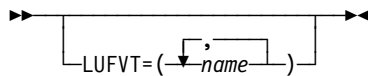
Use: User-written code and Internet Protocol
Default: None

Specifies the names of up to nine function vector tables (FVTs) to be associated with this virtual logical unit. LUFVT is valid only if you code VIRTUAL=YES on the GROUP definition statement.

If this NCP includes virtual resources, specify at least one name to identify an FVT for the SSCP-LU session. You can specify additional names to identify FVTs for LU-LU sessions for this resource.

If you specify NUMSESS, code two names: (1) the name of the FVT for the NAU control block and (2) the name of the FVT for the session control blocks. If you omit NUMSESS, you can specify one to nine names.

The names you specify for LUFVT are related by position to the names you specify for the LUCB keyword. For each name you specify for LUFVT, you must define a corresponding name in the same relative position for LUCB.

Include an EXTRN statement for each unique FVT name in the user source code that is copied into the tables during assembly. (See SRCHI and SRCLO on the GENEND definition statement on page 2-93.)

FVTs must be preassembled and included in the user object modules. (See INCHI on the GENEND definition statement on page 2-96.)

Internet Protocol: When NCST=IP is specified on the GROUP definition statement, LUFVT=(CXSXFVT,CXSXFVT) is the only valid value. If you do not code LUFVT, NDF adds LUFVT=(CXSXFVT,CXSXFVT) to the generation definition.

LUNTFY

```
                  ┌─LUNTFY=1───────┐
►►──┼────────────────┼──►◄
                  └─LUNTFY=─┬─1─┬──┘
                             ├─2─┤
                             └─3─┘
```
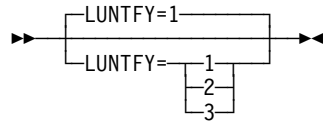
Use: User-written code
Default: 1

Specifies the use of the notify byte in user-written control blocks and whether control is to be passed to a user-written notify task. LUNTFY is valid only if you specify VIRTUAL=YES on the GROUP definition statement.

1

Sets and resets the virtual route blocked bit in the notify byte of the programmed resource logical unit block (NLB) or the programmed resource logical unit block extension (NLX); passes control to the notify task.

2

Sets and resets the virtual route blocked bit in the notify byte of the NLB or NLX; does not pass control to the notify task.

3

Sets and resets the virtual route blocked bit in the programmed resource virtual line block (VLB) and in the NLB or NLX; does not pass control to the notify task.

MAXCOLL
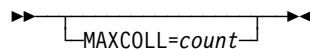
```
►►──┬──────────────────┬──►◄
    └─MAXCOLL=count─┘
```

Default: See text
Minimum: 1
Maximum: 65535

Specifies the maximum number of resources on which data can be collected at one time by NPM. MAXCOLL is coded for logical units defined within the NPM programmed resource group (NPARSC=YES on the GROUP definition statement). See the *NCP, SSP, and EP Resource Definition Guide* for more information.

The default is the number of resources eligible for data collection. This is the number of resources defined with NPA=YES plus the number of resources in the logical unit dynamic reconfiguration pool (LUDRPOOL) if dynamically reconfigured resources are eligible for data collection (NPA=(YES,DR) on the BUILD definition statement).

NCP storage is reserved for NPM data collection based on the value of MAXCOLL. NCP storage is wasted if the value of MAXCOLL is greater than necessary. This value should be no greater than the number of resources eligible for collection in this NCP.

3746 Model 900 Token Ring: 3746 Model 900 Token-Ring resources do not use the storage reserved by MAXCOLL. Therefore, do not include these resources when calculating the value of MAXCOLL.

MAXSESS

```
           ┌─MAXSESS=65535─┐
►►─────────┼───────────────┼───►◄
           └─MAXSESS=sessions─┘
```
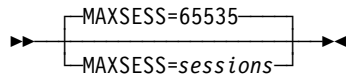
Use: Independent logical units
Default: See text
Minimum: 1
Maximum: 65535

Specifies how many LU-LU sessions this boundary logical unit can have in this link station. MAXSESS is valid only for independent logical units (LOCADDR=0 on this LU definition statement and PUTYPE=2 and XID=YES on the PU definition statement for this LU).

The default is the value of MAXSESS on the BUILD definition statement, which specifies the maximum number of LU-LU sessions for all independent logical units in this NCP. If you do not code MAXSESS on the BUILD definition statement, the default is 65535.

Notes:

1. If this independent logical unit is able to establish sessions over more than one link station, the independent logical unit can have the number of sessions defined by MAXSESS=*sessions* on each link station.

2. For MAXSESS to be valid, the GROUP definition statement must be coded with:

   - LNCTL=SDLC or LNCTL=CA
   - VIRTUAL=NO
   - CHANLNK=NO

MAXTP
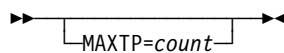
```
►►──┬─────────────────┬──►◄
    └─MAXTP=count─┘
```

Default: See text
Minimum: 1
Maximum: 65535

Specifies the maximum number of resources on which data can be collected by transmission priority at one time by NPM. MAXTP is valid only for logical units defined within the NPM programmed resource group (a group for which NPARSC=YES is specified on the GROUP definition statement). If both MAXTP and MAXCOLL are coded, the value coded on the MAXTP keyword cannot exceed

the value coded on the MAXCOLL keyword; otherwise, the generation definition will fail.

If you code NPA=YES or NPA=(YES,DR) on the BUILD definition statement, the default for MAXTP is either the value coded on MAXCOLL on this LU definition statement or the number of resources defined with NPATP=YES, whichever is less. If you code NPA=(YES,DRTP) on the BUILD definition statement, the default for MAXTP is either the value coded on MAXCOLL on this LU definition statement or the number of resources defined with NPATP=YES plus the value of NUMBER on the PUDRPOOL definition statement, whichever is less.

Specify a value as close as possible to the actual maximum number of resources on which data is to be collected. Specifying a larger value wastes NCP storage.

NETID

```
►►──────────────────────►◄
     └─NETID=name─┘
```
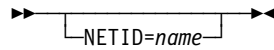
Use: Independent logical units
Default: None

Specifies the name of the network that contains this logical unit. NETID is valid only for independent logical units (LOCADDR=0 on this LU definition statement and PUTYPE=2 and XID=YES on the PU definition statement for this LU). This network name may be different from NETID on the BUILD or PU definition statement.

NEXT

```
►►──────────────────────────────────►◄
     │                    ┌─D─┐      │
     └─NEXT=(nextlabel,───┼───┼──)───┘
                          ├─D─┤
                          └─H─┘
```

Specifies a convention for constructing unique labels for the definition statements that will be copied when you code the AUTOCOPY keyword on the LINE definition statement. NEXT is valid only when you code AUTOCOPY on the previous LINE definition statement. NEXT is required when you code AUTOCOPY on the LINE definition statement.

See AUTOCOPY on the LINE definition statement on page 2-196 for more information. See the *NCP, SSP, and EP Resource Definition Guide* for more information about using NEXT.

nextlabel

Default: None

Specifies the label for the first duplicated copy of this LU definition statement. The label consists of 1 to 8 characters; the first character must be an alphabetical character from A to Z, #, or @. NDF uses this *nextlabel* to determine what positions in the label to use for incrementing label names and the amount of the increment. The value you enter for *nextlabel* must be the same length as the statement label, and must differ from the label in such a way that NDF can identify a numeric field to use for incrementing. You can use only one field for

incrementing. This field can increase in length as it is incremented, but it cannot increase so much that the total label length becomes greater than 8 characters.

The format of the statement *label* is

`<p>lo_num_field<s>`

The format of the label given to the first copy of this LU definition statement is

`<p>next_num_field<s>`

where:

- `p` is the prefix of the numeric field and must be the same for both `label` and `nextlabel`.

- `s` is the suffix of the numeric field and must be the same for both `label` and `nextlabel`.

NDF subtracts `lo_num_field` from `next_num_field` to obtain the value it uses to increment the numeric field for the next label.

The numeric fields cannot differ in length, but must differ in value. `next_num_field` must be greater than `lo_num_field`.

If NDF cannot determine how to increment a label, it issues an error message.

D|H

Default: D

Specifies whether the increment and the field to increment are decimal (D) or hexadecimal (H). What you specify here does not need to match what you code for AUTOCOPY or for NEXT on other resource definition statements being replicated.

NPACOLL

```
►►──┬─────────────────────┬──►◄
    └─NPACOLL=─┬─YES─┬─────┘
              └─NO──┘
```

Default: See text

Specifies whether performance data for this logical unit is eligible for collection by the network performance analyzer. NPACOLL=YES is valid only if you code NPACOLL=YES on the PU definition statement or VIRTUAL=YES on the GROUP definition statement. The default is YES if NPACOLL=YES on the PU definition statement; otherwise, the default is NO.

If you code LNCTL=CA on GROUP and NPACOLL=YES on the PU definition statement, NPACOLL specifies that the data collected is from the channel adapter traffic.

If you code NPA=(YES,DR) on the BUILD definition statement and LUDR=YES on LU, the value you define for NPACOLL is ignored. NCP uses NPACOLL=YES.

If you code NPARSC=YES on the GROUP definition statement, the logical units defined within this group are the only resources that can collect NPA data.

If you code GENILU=NO on the BUILD definition statement, NCP collects data for independent logical units for which LUDR=YES is specified only if NPA=(YES,DR) or NPA=(YES,DRTP) is specified on the BUILD definition statement.

Token Ring: To collect performance data from logical units attached to Token-Ring lines, code NPA=(YES,DR) or NPA=(YES,DRTP) on the BUILD definition statement.

NUMSESS

```
►►──┬─────────────────┬──►◄
    └─NUMSESS=hscbs───┘
```
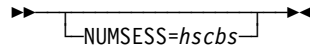
Use:	User-written code
Default:	None
Minimum:	1
Maximum:	255

Specifies the number of half-session control blocks to be associated with the NAU control block for this logical unit. If you omit NUMSESS, specify the names of up to nine half-session control blocks on the LUCB and LUFVT keywords.

The NUMSESS keyword is valid only if VIRTUAL=YES is coded on the GROUP definition statement.

If NCST=IP is coded on the GROUP definition statement, NUMSESS=1 is the only valid value.

PACING

```
            ┌─PACING=(1,,ADAP)─────────────────────────────┐
►►──┼──────────────────────────────────────────────────────┼──►◄
    │              ┌─1──────┐        ┌─ADAP─┐                │
    └─PACING=(─────┼────────┼──,──┬───────┬──,──┼─ADAP──┼──)─┘
                   └─number─┘     └─vtam─┘       └─FIXED─┘
```
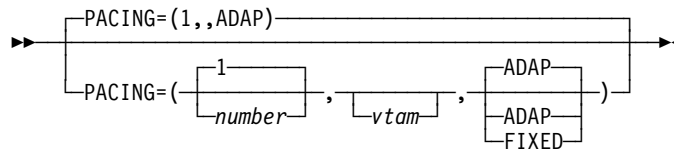
Controls session pacing for the route extension (REX) stage of an LU-LU session. For more information about session pacing, see the *NCP and EP Reference*.

number

Default:	1 for VIRTUAL=NO; 0 for VIRTUAL=YES
Minimum:	0
Maximum:	255

For fixed session pacing, PACING specifies how many PIUs this logical unit can send to NCP before transmission stops to await a pacing response from NCP.

For adaptive session pacing, PACING defines the batch mode window size. The batch mode window size is the number of PIUs, which follow a PIU with a pacing request, that this logical unit can send to NCP before the logical unit must wait for a pacing response from NCP. If you do not code PACING, or if you specify PACING=0, NCP uses an adaptive algorithm to calculate the window size.

vtam
Is an optional VTAM-only numeric value.

ADAP│FIXED

Default: ADAP

Specifies whether NCP is to use adaptive or fixed session pacing. If you specify ADAP (or use ADAP by default) and the logical unit supports adaptive session pacing, NCP uses adaptive session pacing. Otherwise, NCP uses fixed session pacing.

REMLU

```
►►──┬────────────────────────────────────────────┬──►◄
    └─REMLU=(name,─┬──────────┬─,─┬─────────┬─)─┘
                   └─modename─┘   └─SECONLY─┘
```
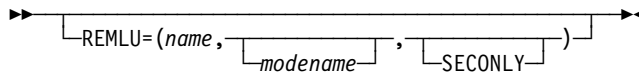
Use: Internet Protocol

Specifies the name of the logical unit partner for this logical unit. When the SSCP-LU session for this NCST logical unit occurs, NCP uses the information you specify on the REMLU keyword to auto-initiate the LU-LU session.

Note: The REMLU keyword is valid only when NCST=IP is coded on the GROUP definition statement.

name

Default: None

Specifies the symbol that identifies the remote logical unit with which the local logical unit will be in session. When the session partner is a remote NCP, specify the identifying statement label coded on the associated NCST LU defi-nition statement in the remote NCP generation definition. When the session partner is an SNALINK application executing in the host, REMLU must match the name of the associated VTAM APPL definition statement in the host VTAM major node definition and the name associated with ACBNAME on that APPL definition statement.

modename

Default: None

Specifies the name of the entry in the log mode table associated with the sec-ondary logical unit. If you do not specify *modename*, the NCST logical unit is defined with the default characteristics established for NCST logical units. For more information about the default characteristics of NCST logical units, refer to the discussion of establishing NCST sessions in the *NCP and EP Reference*.

SECONLY

Default: See text

Specifies that the remote logical unit is a secondary logical unit. When you code SECONLY, the local logical unit is considered the primary logical unit. When you do not code SECONLY, the local logical unit is considered the primary logical unit only if it is higher in collating sequence than the remote logical unit.

RESSCB

```
        ┌─RESSCB=0────────┐
►►──────┤                 ├──►◄
        └─RESSCB=number───┘
```
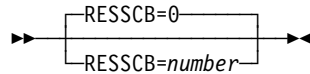
Use: Independent logical units
Default: 0
Minimum: 0
Maximum: Value of MAXSESS on the LU definition statement

Specifies how many BSBs are to be reserved for this independent logical unit.

Independent logical units are coded with LOCADDR=0 on the LU definition statement and PUTYPE=2 and XID=YES on the PU definition statement.

The RESSCB keyword is valid only when VIRTUAL=NO and CHANLNK=NO are coded on the GROUP definition statement.

Note: See the *NCP, SSP, and EP Resource Definition Guide* for more information about defining control block resources. See the control block management information in the *NCP and EP Reference* for detailed information about control blocks that get created for keywords.

UCCB
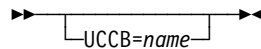
```
►►────────────────────►◄
    └─UCCB=name─┘
```

Use: User-written code
Default: None

Specifies the name of the user channel control block associated with this logical unit if user-written code defines the channel operation. UCCB is required, and only valid, if you code CHANLNK=YES on the GROUP definition statement for this logical unit. Specify a different name for UCCB for each logical unit.

VPACING

```
        ┌─VPACING=(0,,ADAP)──────────────────────────────┐
►►──────┤                                                ├──►◄
        │              ┌─0──────┐        ┌─ADAP─┐        │
        └─VPACING=(────┤        ├──,──┬──────┬──,──┤─ADAP─├──)─┘
                       └─number─┘     └─vtam─┘   └─FIXED─┘
```
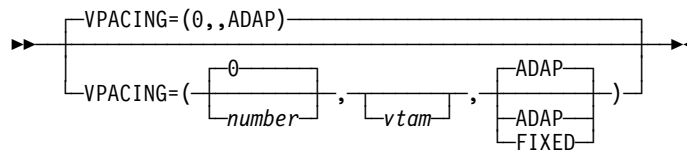
Controls session pacing for the subarea side of an LU-LU session. For more information about session pacing, see the *NCP and EP Reference*.

Note: VPACING is not valid when LEVEL5=USER is coded on the GROUP definition statement.

number

   Default: 0
   Minimum: 0
   Maximum: 63

For fixed session pacing, VPACING specifies how many PIUs this logical unit can send to NCP before transmission stops to await a pacing response from NCP.

For adaptive session pacing, VPACING defines the batch mode window size. The batch mode window size is the number of PIUs, which follow a PIU with a pacing request, that this logical unit can send to NCP before the logical unit must wait for a pacing response from NCP. If you do not code VPACING, or if you specify VPACING=0, NCP uses an adaptive algorithm to calculate the window size.

vtam
Is an optional VTAM-only numeric value.

ADAP│FIXED

Default: ADAP

Specifies whether NCP is to use adaptive or fixed session pacing. If you specify ADAP (or use ADAP by default) and the logical unit supports adaptive session pacing, NCP uses adaptive session pacing. Otherwise, NCP uses fixed session pacing.

LUDRPOOL Definition Statement

Usage: Dynamic reconfiguration or switched SDLC peripheral telecommunication links.

Token Ring: If you include Token-Ring peripheral resources in this generation definition, you must specify the LUDRPOOL definition statement.

Note: Coding large values for the keywords on the LUDRPOOL definition statement requires large amounts of NCP storage.

Position: Must precede first GROUP definition statement.

Keywords: Table 2-85 shows the keywords described under this definition statement.

*Table 2-85. Keywords on the LUDRPOOL Definition Statement*

Keyword	Page	Required	Sift	Default	Tuning*
NUMILU	2-311			0	
NUMTYP1	2-312			0	
NUMTYP2	2-313			0	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►─┬────────┬─►◄
   └─symbol─┘
```
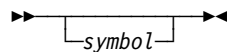
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

NUMILU
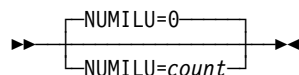
```
         ┌─NUMILU=0─────┐
►►──────┼──────────────┼──►◄
         └─NUMILU=count─┘
```

Default: 0
Minimum: 0
Maximum: See text

Specifies the number of independent logical units for dynamic reconfiguration. When setting up parallel sessions, include room for control session overhead (for example, the SNA service manager) in the value of NUMILU. These sessions can exist between type 2 physical unit nodes.

If your communication controller has 8MB of storage or less, the maximum value for NUMILU is 32 765. If your communication controller has more than 8MB of storage, the maximum value for NUMILU is 65 532. The sum of NUMILU plus NUMTYP1 plus NUMTYP2 must not exceed 65 532.

When GENILU=NO is specified on the BUILD definition statement, NDF increases the value of NUMILU by 1 for each independent logical unit with LUDR=YES in your generation definition.

Notes:

1. You can define part of the NCP buffer pool to be used for dynamically created independent logical unit resources when explicitly defined resources have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* and the *NCP and EP Reference* for more information about dynamically created control blocks.

2. The NUMILU, NUMTYP1, and NUMTYP2 keywords each create control blocks in several control block pools. Some of these control blocks are in the same pools, some are not. See the *NCP, SSP, and EP Resource Definition Guide* for more information about the interactions among the control blocks created for NUMILU, NUMTYP1, and NUMTYP2. For details about which control blocks each keyword creates, see the control block management information in the *NCP and EP Reference*.

NUMTYP1

```
              ┌─NUMTYP1=0─────┐
►►────────────┼───────────────┼──────►◄
              └─NUMTYP1=count─┘
```

Default: 0
Minimum: 0
Maximum: See text

Specifies how many logical units the dynamic reconfiguration pool is to contain for type 1 physical units. To conserve controller storage, base the value of NUMTYP1 on the anticipated dynamic resource requirements, not the maximum number of resources.

If your communication controller has 8MB of storage or less, the maximum value for NUMTYP1 is 24 573, and the sum of NUMTYP1 plus NUMTYP2 must not exceed 24 573. If your communication controller has more than 8MB of storage, the maximum value for NUMTYP1 is 65 532, although the sum of NUMILU plus NUMTYP1 plus NUMTYP2 must not exceed 65 532.

Notes:

1. You can define part of the NCP buffer pool to be used for dynamically created independent logical unit resources for type 1 physical units when explicitly defined resources have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* and the *NCP and EP Reference* for more information about dynamically created control blocks.

2. The NUMILU, NUMTYP1, and NUMTYP2 keywords each create control blocks in several control block pools. Some of these control blocks are in the same pools, some are not. See the *NCP, SSP, and EP Resource Definition Guide* for more information about the interactions among the control blocks created for NUMILU, NUMTYP1, and NUMTYP2. For details about which control blocks each keyword creates, see the control block management information in the *NCP and EP Reference*.

NUMTYP2

```
                  ┌─NUMTYP2=0──────┐
►►─────────────────┼────────────────┼──────────►◄
                  └─NUMTYP2=count──┘
```

Default: 0
Minimum: 0
Maximum: See text

Specifies how many dependent logical units the dynamic reconfiguration pool is to contain for type 2 and type 2.1 physical units. To conserve controller storage, base the value of NUMTYP2 on the anticipated dynamic resource requirements, not the maximum number of resources.

If your communication controller has 8MB of storage or less, the maximum value for NUMTYP2 is 24 573, and the sum of NUMTYP1 plus NUMTYP2 must not exceed 24 573. If your communication controller has more than 8MB of storage, the maximum value for NUMTYP2 is 65 532, although the sum of NUMILU plus NUMTYP1 plus NUMTYP2 must not exceed 65 532.

Notes:

1. You can define part of the NCP buffer pool to be used for dynamically created independent logical unit resources for type 2 and type 2.1 physical units when explicitly defined resources have been exhausted. For more information, see DYNPOOL on the BUILD definition statement on page 2-19. See the *NCP, SSP, and EP Resource Definition Guide* and the *NCP and EP Reference* for more information about dynamically created control blocks.

2. The NUMILU, NUMTYP1, and NUMTYP2 keywords each create control blocks in several control block pools. Some of these control blocks are in the same pools, some are not. See the *NCP, SSP, and EP Resource Definition Guide* for more information about the interactions among the control blocks created for NUMILU, NUMTYP1, and NUMTYP2. For details about which control blocks each keyword creates, see the control block management information in the *NCP and EP Reference*.

LUPOOL Definition Statement

Usage: SDLC devices attached to switched peripheral telecommunication links. One for each access method that communicates with type 1 and type 2 physical units over switched SDLC links using the Assign Network Address (ANA) command. Do not code LUPOOL for any access method that supports Request Network Address Assignment (RNAA) instead of ANA.

Position: Must precede first GROUP definition statement

Keywords: Table 2-86 shows the keyword described under this definition statement.

Table 2-86. Keywords on the LUPOOL Definition Statement

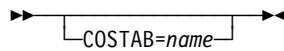| Keyword	Page	Required	Sift	Default	Tuning*
NUMBER	2-315	Required			

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

TCAM: TCAM supports both RNAA and ANA. Only TCAM V2R4 and earlier releases support the LUPOOL definition statement.

VTAM Keyword: The following diagram shows the VTAM keyword that can be coded on this definition statement. This keyword provides information to the VTAM initialization process. It is not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for a description of this keyword and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

Other VTAM keywords can be specified by coding VTM. as a prefix in front of the VTAM keyword.

```
►►─┬─────────────────────────┬─►◄
   └─OWNER=owner_name─┘
```

Keyword Descriptions

symbol

```
►►──symbol──►◄
```

Use: Required when you code the LUPOOL definition statement
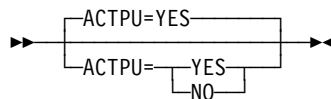Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

NUMBER

►►──NUMBER=*count*──►◄

Use: Required when you code the LUPOOL definition statement
Default: None
Minimum: 1
Maximum: 16381

Specifies the number of logical units to be included in the logical unit pool.

To determine the highest value you may code for NUMBER, assume that (1) NCP is communicating simultaneously with type 1 and type 2 physical units over all of the switched links associated with this access method, and (2) the physical units involved are those with the most logical units. The total number of logical units in all connected physical units represents the greatest possible demand that can be placed on the LUB pool. By specifying this value, you can be sure that the pool will never be depleted.

The value of NUMBER can be less than the total number of logical units in all connected physical units because the previous assumptions represent extreme conditions that rarely occur. However, specifying a very low value can deplete the pool. After a physical connection is established, if NCP is unable to supply sufficient LUBs from the pool, NCP will break the physical connection before transmission begins. NCP will then make a new connection and attempt LUB allocation again. Occasional occurrences of this type can be tolerated, but frequent occurrences indicate the need to increase the size of the pool.

MTALCST Definition Statement

Usage: MTA telecommunication links. One definition statement for each line control selection table (LCST) entry defined.

Position: All MTALCST definition statements must appear together for an MTALCST group. MTALCST must precede the first GROUP definition statement.

Keywords: Table 2-87 shows the keywords described under this definition statement.

Table 2-87. Keywords on the MTALCST Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ACR	2-317			NO	
CLOCKNG	2-317			INT	
CODE	2-317				
COMPARE	2-318			01	
CRRATE	2-319				
DATRATE	2-319			LOW	
GROUP	2-320	Yes			
LCTYPE	2-320			1050	
LINESIZ	2-321				
MASK	2-321			31	
RETRIES	2-321			2	
SPEED	2-322				
TRANSFR	2-322				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►——*symbol*——►◄

Use: Required when you code the MTALCST definition statement
Default: None

Provides a name for the line control selection entry table defined by the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

The LCST keyword on the MTATABL definition statement refers to *symbol*.

ACR

```
        ┌─ACR=NO───────┐
▶▶──────┼──────────────┼──▶◀
        └─ACR=──┬─YES─┬┘
                └─NO──┘
```
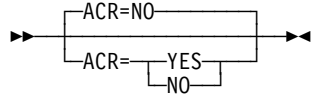
Use: IBM 1050
Default: NO

Specifies whether the IBM 1050 Data Communication System terminals that can call the controller on this line are equipped with the accelerated carriage return feature. Code ACR=YES only if all IBM 1050 terminals that can call the controller over this line are equipped with the feature.

CLOCKNG

```
    ┌─CLOCKNG=INT───────┐
▶▶──┼───────────────────┼──▶◀
    └─CLOCKNG=──┬─INT─┬──┘
               └─EXT─┘
```
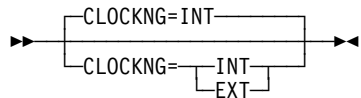
Default: INT

Specifies whether the communication scanner or the modem (data set) provides clocking. Your system designer can provide you with this information.

If the scanner provides clocking, code CLOCKNG=INT. If the modem provides clocking, code CLOCKNG=EXT.

Note: Be sure to spell CLOCKNG without the letter *I*.

CODE

```
▶▶──┬──────────────────┬──▶◀
    └─CODE=──┬─ASCII─┬──┘
             ├─BCD───┤
             ├─BCD2──┤
             ├─COR───┤
             ├─COR2──┤
             ├─DIC1──┤
             ├─DIC3──┤
             └─EBCD──┘
```
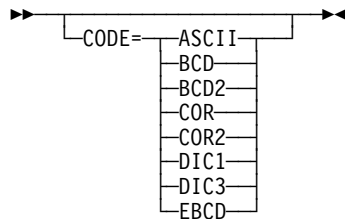
Specifies the transmission code to be used to communicate with the type of terminal represented by this definition statement. Associated with each transmission code is a translation table in NCP. The contents of the translation tables are shown in the "Line Character Codes" section of *NCP and EP Reference Summary and Data Areas*, Volume 2.

Table 2-88 on page 2-318 shows the valid and default transmission codes for each type of terminal.

Table   2-88. Values for CODE on the MTALCST Definition Statement

Transmission Code	CODE=	BSC Devices	IBM 1050	IBM 2740	IBM 2741	TWX	WTTY
ASCII	ASCII					▪	
Binary Coded Decimal 1	BCD		▪	▪	▪		
Binary Coded Decimal 2	BCD2		▪		▪		
Correspondence 1	COR			▪	D		
Correspondence 2	COR2				▪		
Data Interchange Code 1	DIC1					D	
Data Interchange Code 3	DIC3					▪	
Extended Binary Coded Decimal	EBCD		D	D	▪		

▪   May be coded
D   Default value

COMPARE

```
                ┌─COMPARE=01──────┐
►►──────────────┤                 ├──►◄
                └─COMPARE=chars───┘
```
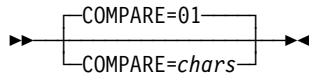
Use: TWX terminals
Default: 01

Specifies the hexadecimal representation of the compare character used during the
LCST entry selection process. The mask character specified by MASK is logically
ANDed with the first character received from the terminal. If the result equals the
compare character specified by COMPARE, the contents of the LCST entry defined
by this MTALCST definition statement are used to initialize the operating character-
istics for the TWX terminal.

COMPARE is not valid when LCTYPE=3767 on this MTALCST definition statement.

chars should be specified as 2 hexadecimal digits. Table 2-89 on page 2-319 con-
tains recommended pairs of values for the COMPARE and MASK keywords.

The COMPARE default and MASK default properly identify the WRU character
transmitted at 110 bps.

Table 2-89. Values for COMPARE on the MTALCST Definition Statement

TWX Terminal Speed	First Received Character	COMPARE=	MASK=
110 bps	@[1]	00	7F
	N	1E	7F
	0[1]	60	7F
	∧	3E	7F
300 bps	@[1]	78	7B
	N	79	7B
	0[1]	7A	7B
	↑[2]	7B	7B

[1] The letter P is used for emulation mode speed selection and may be desired for compatibility. If the letter P is used for 110 baud, specify COMPARE=00 and MASK=5F, and do not use the characters @ and 0.

[2] ASCII X'5E' character. This character is represented on terminal keyboards as ∧, ¬, or ↑. Do not use this character if an IBM 3767 can call on the same line as a TWX terminal, because NCP cannot distinguish between this character and the initial end-of-address (circle D) character automatically sent by the IBM 3767.

CRRATE

```
►►──┬──────────────────────────┬──►◄
    └─CRRATE=print_positions─┘
```
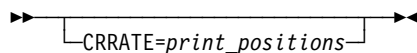
Default: See text
Minimum: 1
Maximum: 255

Specifies how many print positions a printer carriage returns for each idle character NCP sends. This value enables NCP to calculate how many idle characters to send to allow the carriage to return completely.

If you omit CRRATE, NCP assumes a rate of 40 print positions per idle character for TWX terminals, 13 for IBM 1050 terminals equipped with the accelerated carrier return (ACR) feature, and 10 for IBM 2740, 2741, and 1050 terminals without the ACR feature.

DATRATE

```
        ┌─DATRATE=LOW──────┐
►►──┼──────────────────┼──►◄
        └─DATRATE=─┬─LOW──┬─┘
                   └─HIGH─┘
```
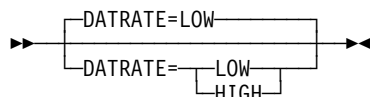
Default: LOW

Specifies the data rate used by the modem (data set) that attaches the multiple-terminal-access line to the controller. DATRATE is valid only if the modem has a dual data rate.

Code DATRATE=HIGH if the high data rate is used. Code DATRATE=LOW or do not code DATRATE if the low data rate is used. If the modem has only one data rate, code DATRATE=LOW or omit DATRATE.

Attention: DATRATE=HIGH is not valid for modems attached to line sets 1A, 1B, 1C, 2A, 3A, 4A, 4B, and 4C. Specifying DATRATE=HIGH can cause a feedback-check error condition.

GROUP

```
►►──GROUP=name──►◄
```

Use: Required when you code the MTALCST definition statement
Default: None

Specifies the name of the stand-alone GROUP definition statement for a line group whose terminal characteristics are the same as those of the terminal that calls the controller over the multiple-terminal-access line. *name* must not be the name of a GROUP definition statement that defines a group of multiple-terminal-access lines.

LCTYPE

```
        ┌─LCTYPE=1050─────┐
►►──────┤                 ├──►◄
        └─LCTYPE=──┬─1050──┬─┘
                   ├─2740A─┤
                   ├─2740D─┤
                   ├─2740E─┤
                   ├─2740F─┤
                   ├─2741──┤
                   ├─3767──┤
                   └─TWX───┘
```
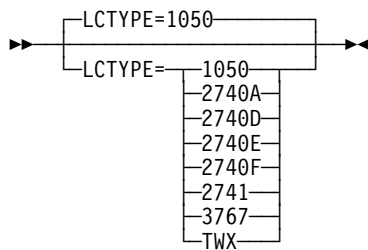
Default: 1050

Specifies the type of terminal and line control used. Use the information in Table 2-90 to determine which entry to code for LCTYPE.

Table 2-90. Values for LCTYPE on the MTALCST Definition Statement

Entry	Type
1050	IBM 1050
2740A	IBM 2740 basic
2740D	IBM 2740 with transmit control feature
2740E	IBM 2740 with transmit control and record checking features
2740F	IBM 2740 with record checking feature
2741	IBM 2741
3767	IBM 3767 in 2741 compatibility mode
TWX	Western Union TWX

LINESIZ

```
►►──┬──────────────────────────────┬──►◄
    └─LINESIZ=print_line_length─┘
```
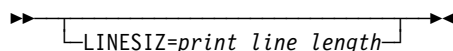
Default: See text
Minimum: 1
Maximum: 255

Specifies the length of the print line for the printer on the line represented by this MTALCST definition statement.

The default is 72 for TWX terminals and 130 for IBM 1050, 2740, and 2741 terminals.

MASK

```
          ┌─MASK=31──────┐
►►──┬──────────────────┬──►◄
    └─MASK=chars──┘
```
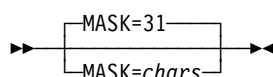
Use: TWX terminals
Default: 31

Specifies the hexadecimal representation of the mask character used during the LCST entry selection process. The mask character is logically ANDed with the first character received from the terminal. If the result equals the compare character specified by the COMPARE keyword, the contents of the LCST entry are used to initialize the operating characteristics for the TWX terminal.

MASK is not valid when LCTYPE=3767 on this MTALCST definition statement.

chars should be specified as 2 hexadecimal digits and must never have a binary 0 specified in the same bit position as the compare character, which is binary 1. Refer to the COMPARE keyword description for a table of recommended mask and compare character pairs.

The MASK default and the COMPARE default properly identify the WRU character transmitted at 110 bps.

RETRIES
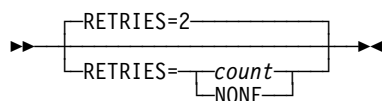
```
          ┌─RETRIES=2───────────┐
►►──┬────────────────────────┬──►◄
    └─RETRIES=──┬─count─┬─┘
               └─NONE──┘
```

Default: 2
Minimum: 1
Maximum: 255

Specifies the maximum number of attempts NCP is to make to recover from text errors in message data sent to or received from an IBM 1050 terminal or an IBM 2740 terminal with record checking. (Other types of multiple-terminal-access terminals cannot retransmit to recover.) *m* is the number of attempts. Any value less than 255 specifies the exact number of attempts; 255 specifies unlimited attempts.

If you code RETRIES=NONE, no error recovery is attempted for read-text or write-text errors.

If you omit RETRIES, a maximum of two recovery attempts are made for terminals that can retransmit. If the terminal cannot retransmit, NONE is assumed and no recovery is attempted.

SPEED

```
►►──┬──────────────┬──►◄
    └─SPEED=rate─┘
```
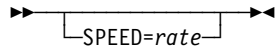
Default: None
Minimum: 1
Maximum: See text

Specifies the data rate (in bits per second) for the multiple-terminal-access lines associated with this LCST entry.

If internal clocking is used (CLOCKNG=INT), code either SPEED=134 or SPEED=300.

If external clocking is used (CLOCKNG=EXT), SPEED is ignored. However, it is recommended that you code this keyword for documentation purposes.

TRANSFR
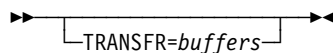
```
►►──┬─────────────────┬──►◄
    └─TRANSFR=buffers─┘
```

Default: See text
Minimum: 1
Maximum: See text

Specifies the maximum number of buffers NCP is to use to receive message text from a terminal before transferring filled buffers to the host processor. If NCP receives the specified number of buffers, it transfers them to the host processor as a subblock. However, NCP continues to receive message text from the terminal until it receives an end-of-block or end-of-transmission character.

The maximum value for TRANSFR, which is also the default, is the smaller of the following values:

- 255

- The value of the following expression, which uses the values of MAXBFRU, UNITSZ, and BFRPAD on the HOST definition statement and BFRS on the BUILD definition statement:

$$((MAXBFRU \times UNITSZ) - BFRPAD) \div BFRS$$

If you do not inhibit subblocking, code a value of 2 or more for TRANSFR.

MTALIST Definition Statement

Usage: MTA telecommunication links. One definition statement for each list of line control types for MTA telecommunication links.

Position: Must follow group of MTALCST definition statements for switched data link or links. Must precede first GROUP definition statement.

Keywords: Table 2-91 shows the keyword described under this definition statement.

Table 2-91. Keywords on the MTALIST Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
LCTYPE	2-323	Yes			

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the MTALIST definition statement
Default: None

Provides a name for the MTA list referred to by the MTALIST keyword of the LINE definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

LCTYPE

►►──LCTYPE=(──↓──*type*──)──►◄

Use: Required when you code the MTALIST definition statement
Default: None

Specifies the terminal types to be included in a list of terminal identification processes. The MTALIST keywords of one or more LINE definition statements for MTA lines can refer to this list. You can specify any of the types shown in Table 2-92 on page 2-324.

Table   2-92. Values for LCTYPE on the MTALIST Definition Statement

Entry	Type
1050	IBM 1050
2740A	IBM 2740 basic
2740D	IBM 2740 with transmit control feature
2740E	IBM 2740 with transmit control and record checking features
2740F	IBM 2740 with record checking feature
2741	IBM 2741
3767	IBM 3767 in 2741 compatibility mode
TWX	Western Union TWX

MTAPOLL Definition Statement

Usage: MTA telecommunication links. One definition statement for all IBM 1050 terminals that can dial the controller over any MTA link.

Position: Must precede first GROUP definition statement.

Keywords: Table 2-93 shows the keyword described under this definition statement.

Table 2-93. Keywords on the MTAPOLL Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
POLL	2-325	Required			

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

POLL

Use: Required when you code the MTAPOLL definition statement
Default: None

Specifies the list of polling characters. Code the hexadecimal representation of the EBCDIC polling characters.

For example, consider a network with two IBM 1050 terminals, each with a keyboard and tape reader as input devices. If the polling characters for the keyboards are A5 and B5 and the polling characters for the tape readers are A6 and B6, you would code POLL=(C1F5,C1F6,C2F5,C2F6).

MTATABL Definition Statement

Usage: MTA telecommunication links. One definition statement for each unique combination of line control types and transmission codes for MTA links.

Position: Must precede first GROUP definition statement.

Keywords: Table 2-94 shows the keywords described under this definition statement.

Table 2-94. Keywords on the MTATABL Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
CODE	2-326				
LCST	2-327				
LCTYPE	2-327			1050	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions
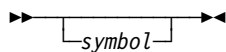
symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

CODE
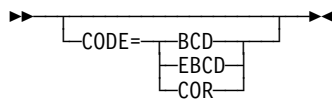
Default: See text

Specifies the transmission code used by the type of terminals represented in this MTA table. Define the same type of code you specify on the MTALCST definition statement named on LCST of this definition statement.

Do not specify CODE if you specify LCTYPE=TWX on this MTATABL definition statement.

Table 2-95 lists the default values for CODE depending on the value of LCTYPE.

Table 2-95. Values for CODE on the MTATABL Definition Statement

LCTYPE=	CODE Default
1050	EBCD
2740	EBCD
2741	COR
3767	COR

LCST

►►──LCST=(──┬──;──┬──)──►◄
            └─*name*─┘

Use: Required when you code the MTATABL definition statement
Default: None

Specifies the name of up to 10 MTALCST definition statements.

Name only those MTALCST definition statements that have the same combination of code and line control as this MTATABL definition statement. This restriction does not apply for TWX line control. You can name any MTALCST definition statement on which LCTYPE=TWX is coded regardless of whether the value of CODE on that definition statement is DIC1, DIC3, or ASCII.

Consider COR2 equivalent to COR, and BCD2 equivalent to BCD for defining LCST. For example, suppose you specify two MTALCST definition statements with LCTYPE=2741, but one with CODE=COR and the other with CODE=COR2. If you specify LCTYPE=2741 and CODE=COR for this MTATABL definition statement, you can name both MTALCST definition statements on LCST.

LCTYPE
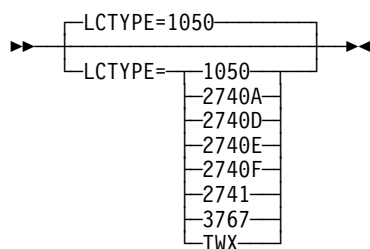
```
        ┌─LCTYPE=1050──────────┐
►►──────┤                      ├──►◄
        └─LCTYPE=──┬─1050──┬───┘
                   ├─2740A─┤
                   ├─2740D─┤
                   ├─2740E─┤
                   ├─2740F─┤
                   ├─2741──┤
                   ├─3767──┤
                   └─TWX───┘
```

Default: 1050

Specifies the type of line control used by the terminals represented in this MTA table. Specify the same line control type you defined on the MTALCST definition statements named by LCST of this MTATABL definition statement.

NCPNAU Definition Statement

Usage: User-written code. One for each user-defined network addressable unit.

Position: Must follow SYSCNTRL definition statement.

Keywords: Table 2-96 shows the keywords described under this definition state-
ment.

Table 2-96. Keywords on the NCPNAU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
NAUCB	2-328				
NAUFVT	2-329	Yes			
NOTIFY	2-329			1	
NUMSESS	2-329				
TYPE	2-330			SSCP	
VIROWNER	2-330				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
NTuneMON V2R5 User's Guide for more information.

Keyword Descriptions

symbol

►►—*symbol*—►◄

Use: Required when you code the NCPNAU definition statement
Default: None

Provides a resource name for the network addressable unit. *symbol* can be any
valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to
avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process
creates.

NAUCB

```
►►────────────────────────────►◄
        │              ┌─,─┐      │
        └─NAUCB=(──▼─name─┘─)─┘
```

Specifies the names of user-defined control blocks associated with this network
addressable unit. The names specified by NAUCB are related by position to the
symbols specified by NAUFVT. If you define a name in NAUCB, specify a corre-
sponding name in the same position in NAUFVT. If you specify NUMSESS, specify
two names. If you omit NUMSESS, specify one to nine names.

NAUFVT

►►──NAUFVT=(─┬─▼─;──┬─*name*─┬──┬──)─►◄

Use: Required when you code the NCPNAU definition statement
Default: None

Specifies the names of the FVTs associated with this network addressable unit. If you specify NUMSESS, you must use two symbols here. The first symbol is the name of the FVT for this NAU. The second symbol is the name of the FVT extension. If you do not specify NUMSESS, specify one to nine symbols. At least one FVT is required. The symbols in NAUFVT are related by position to the symbols in NAUCB.

Include an EXTRN statement for each unique FVT symbol in the user source code that is copied into the tables during assembly. (See SRCHI and SRCLO on the GENEND definition statement on page 2-93 for more information.)

Note: FVTs must be preassembled and included in the user object modules. (See INCHI and INCLO on the GENEND definition statement on page 2-96 for more information.)

NOTIFY

►►──┬─NOTIFY=1──────────┬──►◄
     └─NOTIFY=─┬─1─┬────┘
               └─2─┘

Default: 1

Specifies use of the notify byte in user-written control blocks and whether control is passed to a user-written notify task. NOTIFY=1 sets and resets the virtual route blocked bit in the notify byte of the NLB or NLX control block and passes control to the notify task. NOTIFY=2 sets and resets the virtual route blocked bit but does not pass control to the notify task.

NUMSESS

►►──┬────────────────┬──►◄
     └─NUMSESS=*hscbs*─┘

Default: None
Minimum: 1
Maximum: 255

Specifies how many half-session control blocks are to be associated with the resource being defined. If you code NUMSESS, specify exactly two names for the NAUFVT and NAUCB keywords.

TYPE

```
            ┌─TYPE=SSCP─────────┐
►►──────────┴───────────────────┴──►◄
            └─TYPE=──┬─SSCP──┬──┘
                     └─NCPLU─┘
```
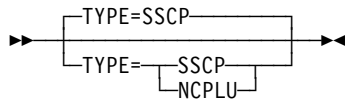
Default: SSCP

Specifies whether this network addressable unit is to function as a system services
control point or as an NCP logical unit.

VIROWNER

```
►►─────┬──────────────────┬──►◄
       └─VIROWNER=name─────┘
```

Specifies the name of the user-written generation application load module that is
responsible for processing the resources defined by this NCPNAU definition state-
ment. This load module must be specified by USERGEN on the OPTIONS defi-
nition statement and must indicate in the start record of its SKVT that it is valid for
VIROWNER.

NETWORK Definition Statement

Usage: SNA network interconnection (SNI). One for each network, other than the native network, that the gateway NCP joins.

Note: The NETWORK definition statement is valid only if you code HSBPOOL on the BUILD definition statement.

Position: Must follow device definition statements (CLUSTER, COMP, LU, PU, and TERMINAL) for native resources attached to the gateway NCP.

Keywords: Table 2-97 shows the keywords described under this definition statement.

Table 2-97. Keywords on the NETWORK Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ACTPU	2-332			YES	
COPIES	2-332				
ERLIMIT	2-333		B		
GWPACING	2-333		B	(0,ADAP,ALLOW)	
MAXSUBA	2-334		B		
NETID	2-335				
NETLIM	2-335		B		
NUMHSAS	2-336		B		
PATHEXT	2-336		B		
SALIMIT	2-337		B	255	
SESSLIM	2-337		B	255	
SUBAREA	2-337	Yes			
TGBXTRA	2-338		B		
VRTIMERn	2-338		B	(0,0,0)	

Sift:

B   BUILD definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements: The values of certain keywords on the BUILD definition statement can sift down to the NETWORK definition statement. These keywords are indicated in Table 2-97. If you code one of these keywords on the BUILD definition statement but not on the NETWORK definition statement, the value for the BUILD definition statement also applies to the NETWORK definition statement. If you do not code the keyword on either definition statement, NCP uses the default value, if any, for both definition statements.

VTAM Keyword: The following diagram shows the VTAM keyword that can be coded on this definition statement. This keyword provides information to the VTAM initialization process. It is not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for a description of this keyword and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

| Other VTAM keywords can be specified by coding VTM. as a prefix in front of the
| VTAM keyword.

```
►►─┬──────────────┬─►◄
   └─COSTAB=name──┘
```

Keyword Descriptions

symbol

```
►►──symbol──►◄
```

Use: Required when you code the NETWORK definition statement
Default: None

Provides a resource name for the network addressable unit. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

ACTPU

```
    ┌─ACTPU=YES───────┐
►►──┤                 ├─►◄
    └─ACTPU=─┬─YES─┬──┘
             └─NO──┘
```

Default: YES

Specifies whether access methods in this non-native network can activate the gateway NCP, that is, whether the gateway NCP can accept ACTPUs from SSCPs in this non-native network.

COPIES

```
►►─┬───────────────┬─►◄
   └─COPIES=count──┘
```
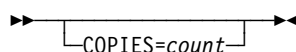
Use: Dynamic network identification
Default: None
Minimum: 1
Maximum: See text

Identifies this NETWORK definition statement and all following GWNAU and PATH definition statements as a model network and specifies how many copies of this

network are to be made available in the network vector table (NVT). The defined GWNAU and PATH definition statements are replicated as defined for each copy of the model network. *count* is the number of entries defined.

A model network uses the following statements:

- A NETWORK definition statement with COPIES coded

- One or more GWNAU definition statements coded with NAME, ELEMENT, and NETID keywords, or one GWNAU definition statement coded with NUMADDR specifying a pool of cross-network resource addresses

- One or more PATH definition statements for the adjacent subarea at the time of dynamic association of NETID. This allows a line using a dynamic entry in the NVT to have a transmission group (TG) attached during link station activation.

After NCP uses each copy and the network is returned to the pool of free networks, the network profile returns to the state in which it was generated.

You can create only one model network. The maximum number of copies of the model network depends on the total number of networks you define; the total cannot exceed 253 non native networks. For example, if you define five explicit networks and one model network, you can create no more than 247 copies of the model network (5 explicit networks + 1 model network + 247 copies = 253 networks). Note that if you code the model network statement last in the generation, you can have one more network (254 total).

You must code either COPIES or NETID on the NETWORK definition statement. When you code COPIES, you must code at least one GROUP definition statement with NETID=(*name*,ANY) and define a subarea link with PUTYPE=4.

Attention: Do not code both NETID and COPIES on the NETWORK definition statement. Your system generation will fail if you code both.

ERLIMIT
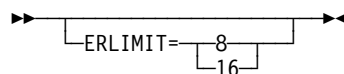
```
►►──┬─────────────────────┬──►◄
    └─ERLIMIT=─┬─8──┬──────┘
              └─16─┘
```

Default: See text

Specifies whether NCP is to support 8 or 16 explicit routes for this non-native network. To save storage, specify 8 explicit routes instead of 16.

Migration Aid Function: You can use the migration aid function to add a value for ERLIMIT to all of the NETWORK definition statements in the generation definition. See NERLIM on the OPTIONS definition statement on page 2-344 for details.

GWPACING
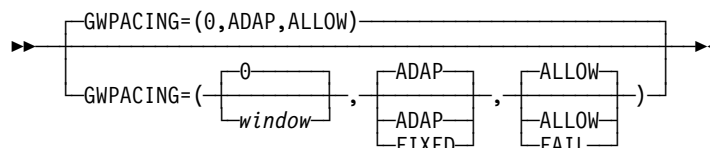
```
       ┌─GWPACING=(0,ADAP,ALLOW)─────────────────────────┐
►►─────┤                                                 ├──►◄
       └─GWPACING=(─┬─0──────┬─,─┬─ADAP──┬─,─┬─ALLOW─┬─)─┘
                    └─window─┘   ├─ADAP──┤   ├─ALLOW─┤
                                 └─FIXED─┘   └─FAIL──┘
```

Use: Gateway NCP
Example: GWPACING=(10,,FAIL)

Specifies the actions to take during a cross-network session if the BIND or response (BIND) indicates that the session will run unpaced (that is, no session pacing will control the flow of data for this session).

window

Default: 0
Minimum: 0
Maximum: 63

Specifies the pacing window size to be used to override the session pacing parameters in a negotiable BIND for SNA LU-LU cross-network sessions. The session pacing parameters on a negotiable BIND are overridden if they indicate that no session pacing is to be used on this session. If you code *window* or allow it to default to zero, the pacing parameters on negotiable BIND are not overridden.

ADAP│**FIXED**

Default: ADAP

Specifies the action to be taken on BIND for SNA LU-LU cross-network sessions for which adaptive session pacing may be used. ADAP allows the BIND to pass through unchanged. FIXED turns off the adaptive session pacing support bit in the BIND, which prevents adaptive session pacing from being used for this session.

ALLOW│**FAIL**

Default: ALLOW

Specifies the action to be taken on nonnegotiable BINDs and negotiable BIND responses for the activation of SNA LU-LU cross-network sessions for which no session pacing will be used. ALLOW allows the session activation attempt to proceed. FAIL prevents the session activation from proceeding.

MAXSUBA
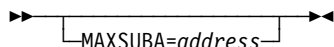
```
►►──┬──────────────────┬──►◄
    └─MAXSUBA=address─┘
```

Default: None
Minimum: 1
Maximum: 255

Specifies the upper limit of the range of subarea addresses used within this non-native network. You can specify a different limit for each network, but all nodes within a network should have the same limit.

The valid values for MAXSUBA are 3, 7, 15, 31, 63, 127, and 255. If you specify any other value less than 255, NCP uses the next higher value.

Code MAXSUBA if this network contains a node that does not support extended network addressing.

NETID

```
►►──┬─────────────┬──►◄
    └─NETID=name──┘
```
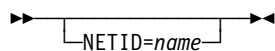
Use: Required when you code the NETWORK definition statement unless
             you code COPIES
Default: None

Defines the name of this non-native network. This name is used in communication between the gateway SSCP and the gateway NCP to address a specific adjacent network represented in the gateway NCP. The maximum length of *name* is 8 characters.

Attention: Do not code both NETID and COPIES on the NETWORK definition statement. Your system generation will fail if you code both.

For proper operation of the gateway NCP, every network in an interconnected network must use the same name to refer to a given network. Although this name is required for proper operation, system generation cannot enforce it.

NETLIM
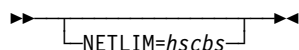
```
►►──┬──────────────┬──►◄
    └─NETLIM=hscbs─┘
```

Minimum: 0
Maximum: 65535

Specifies how many HSCBs can be assigned at one time to all addresses in this non-native network. This limit includes HSCBs that are assigned to both predefined and nonpredefined resources. This limit prevents a single network from using all the HSCBs in the HSCB pool.

Note: Even if DYNPOOL is coded, NCP will not allow the number of HSCBs to exceed the value coded for NETLIM.

If you code SESSLIM=0 on this NETWORK definition statement, the default value of NETLIM is 0 since no HSCBs can be taken from the pool. Otherwise, the default is 5000.

If the value of NUMSESS on the GWNAU definition statement is less than the value of SESSLIM, the gateway NCP allocates additional control blocks from the pool to the SSCP or logical unit until the number of control blocks taken from the pool reaches the value of NETLIM or SESSLIM, whichever is lower. Remember that NUMSESS defines a permanent assignment for a specific SSCP or logical unit and that SESSLIM defines a pool of control blocks to be assigned dynamically.

If the value of NUMSESS is greater than or equal to the value of SESSLIM, the gateway NCP allocates the requested number of control blocks, but does not allocate additional control blocks from the pool, even if needed.

NUMHSAS

```
►►──────────────────────────►◄
      └─NUMHSAS=subareas─┘
```
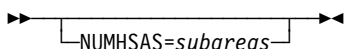
Default: None
Minimum: 0
Maximum: See text

Specifies the expected number of subareas in this non-native network that have virtual routes ending in this NCP. This number gets added to the value of the NUMHSAS keyword on the BUILD definition statement. NCP does not keep a separate resource pool for each network.

The maximum value for NUMHSAS is equal to the value of SALIMIT on this NETWORK definition statement or 65534, whichever value is less.

See NUMHSAS on the BUILD definition statement on page 2-44 for more information.

PATHEXT

```
►►──────────────────────────►◄
      └─PATHEXT=entries─┘
```
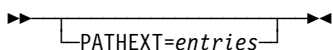
Use: Dynamic update facility
Default: See text
Minimum: 0
Maximum: 65534

Specifies how many extra TRT entries generated for this non-native network are to be available for added destination subareas.

The default is 254 minus the number of destination subareas defined for this network.

A TRT row is generated for each unique destination subarea defined by a PATH definition statement. PATHEXT specifies how many extra TRT rows are to be generated for this network after the PATH definition statements for this network are processed. You need one extra row for each PATH definition statement added through dynamic path update. If you do not code PATHEXT, NCP uses a default for the number of TRT rows generated for this network.

If you code an excessive value for PATHEXT, you could waste NCP storage. The sum of TRT rows generated for this network may not exceed the value of SALIMIT on the NETWORK definition statement, minus 1. The sum of all TRT rows in the generation may not exceed 65 535.

SALIMIT
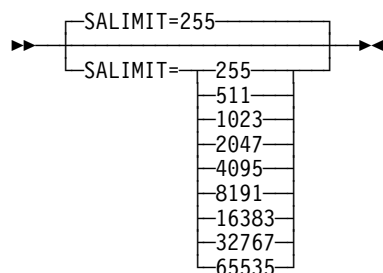
```
                  ┌─SALIMIT=255─┐
►►────────────────┼─────────────┼────────────►◄
                  └─SALIMIT=──┬─255───┬─┘
                              ├─511───┤
                              ├─1023──┤
                              ├─2047──┤
                              ├─4095──┤
                              ├─8191──┤
                              ├─16383─┤
                              ├─32767─┤
                              └─65535─┘
```

Default: 255
Minimum: 0
Maximum: 65535

Specifies the largest subarea address supported by this node in the non-native network.

Valid values are 255, 511, 1023, 2047, 4095, 8191, 16383, 32767, and 65535.

SESSLIM

```
             ┌─SESSLIM=255───┐
►►───────────┼───────────────┼──►◄
             └─SESSLIM=hscbs─┘
```
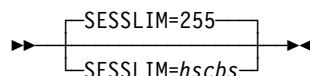
Default: 255
Minimum: 0
Maximum: 255

Specifies the maximum number of HSCBs that can be assigned to an address in this non-native network. Although the address resides in the non-native network, the SSCP or logical unit that it represents resides in another network.

In some cases, SESSLIM also specifies the maximum number of sessions in which a predefined resource can participate. See NETLIM on the NETWORK definition statement on page 2-335 for more information.

SUBAREA

```
►►───SUBAREA=address───►◄
```
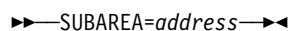
Use: Required when you code the NETWORK definition statement
Default: None
Maximum: Value of SALIMIT on this NETWORK definition statement.

Specifies the subarea address of the gateway NCP as it is known in this non-native network. (SUBAREA on the BUILD definition statement specifies the subarea address of the gateway NCP as it is known in the native network.)

The gateway NCP can be known by the same subarea address in each network that it joins. However, its address in any particular network must be unique.

TGBXTRA

```
►►──┬────────────────┬──►◄
     └─TGBXTRA=tgbs─┘
```
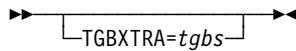
Default: See text
Minimum: 0
Maximum: 65534

Specifies how many extra TGBs are to be generated after processing all PATH definition statements for the native network.

The default value for TGBXTRA is the number of subarea links and subarea channels in NCP minus the number of unique adjacent subarea and transmission group number (*adjsa,tgn*) pairs defined by the PATH definition statements for this network.

If you code an excessive value for TGBXTRA, you could waste NCP storage.

VRTIMERn

```
         ┌─VRTIMERn=(0,0,0)──────────────────────────────┐
►►──┼───────────────────────────────────────────────────┼──►◄
     │                 ┌─0──┐      ┌─0──┐      ┌─0──────┐  │
     └─VRTIMERn=(──┬────┬──,──┬────┬──,──┬────────┬──)─┘
                      └─alert─┘      └─deact─┘      └─queued─┘
```
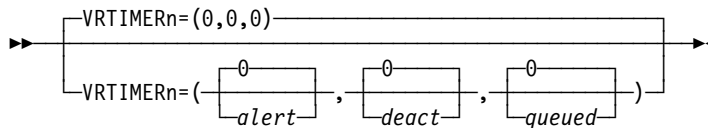
Specifies the conditions under which NCP is to send a blocked virtual route alert to the NetView program or deactivate a held virtual route in this non-native network. For more information on blocked virtual route alerts, refer to the *NCP and EP Reference*.

In VRTIMER*n*, *n* can be 0, 1, or 2. VRTIMER0, VRTIMER1, and VRTIMER2 indicate timer values applied to virtual routes with a transmission priority of 0, 1, and 2, respectively.

alert

Default: 0
Minimum: 0
Maximum: 86400

Specifies, in seconds, the virtual route congestion limit. If a virtual route is congested for this length of time, NCP sends a blocked virtual route alert to the NetView program. NCP sends the alert even if the virtual route pacing response is received during the time the virtual route is congested. If the value of *alert* is 0, NCP does not send blocked virtual route alerts.

If the value you code is not an even multiple of 10, NDF rounds the value up to the nearest multiple of 10. The maximum of 86400 seconds equals one day.

deact

Default:	0
Minimum:	0
Maximum:	86400

Specifies, in seconds, the held virtual route limit. If a virtual route is held for this length of time, NCP deactivates the route and sends an alert to the NetView program. If the value of *deact* is 0, NCP does not deactivate virtual routes because of a held condition.

If you code a value other than 0 for *deact*, that value must be greater than or equal to *alert*. If the value you code is not an even multiple of 10, NDF rounds the value up to the nearest multiple of 10. The maximum of 86400 seconds equals one day.

queued

Default:	0
Minimum:	0
Maximum:	255

Specifies the number of PIUs that constitutes congestion. If the number of PIUs on a virtual route's transmit queue is equal to or greater than *queued*, the virtual route is congested. If the value of *queued* is 0, NCP does not send alerts for congested virtual routes.

If you code 0 for *alert*, you must also code 0 for *queued*.

OPTIONS Definition Statement

Usage: Used to include or exclude any of the following options:

- Verifying a generation definition without building tables
- Collecting a variety of NDF trace information
- Loading a generation application load module that uses the NDF standard attachment facility (not valid for EP standalone)
- Creating a new generation definition
- Using the migration aid function

Position: If you specify the USERGEN, NEWDEFN, or FASTRUN keyword, the OPTIONS definition statement must be the first definition statement in the generation definition. Otherwise, the OPTIONS definition statement can occur anywhere in the generation definition.

Keywords: Table 2-98 shows the keywords described under this definition statement.

Table 2-98 (Page 1 of 2). Keywords on the OPTIONS Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ADDRCONV	2-342				
CHANNELS	2-342			GROUP	
DPU	2-343			YES	
FASTRUN	2-343				
NDNAME	2-343				
NERLIM	2-344				
NEWDEFN	2-344				
NOTRDATA	2-345				
NOTRGLOB	2-346				
NOTRPARM	2-346				
NOTRPROC	2-347				
REMOVCOM	2-347			NO	
SAVEADDR	2-347				
TMODEL	2-348				
TOCONV	2-349				
TRDATA	2-349				
TRGLOB	2-350				
TRPARM	2-350				
TRPROC	2-350				
TRSNAP	2-351				
TUSGTIER	2-351				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-98 (Page 2 of 2). Keywords on the OPTIONS Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
TVERSION	2-352				
USERGEN	2-352				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

NDF Subcomponents for Trace Keywords: You can specify the following NDF subcomponents for the TRDATA, NOTRDATA, TRGLOB, NOTRGLOB, TRPARM, NOTRPARM, TRPROC, and NOTRPROC keywords:

ICNAL[1,2]	Automatic line generation
ICNCR[1]	Cross reference
ICNCV[1]	Consistency and validity checking
ICNER[1]	Error processor
ICNIO	I/O
ICNIP	Input processor
ICNLE	Link-edit
ICNMN	Main processor
ICNOB	Object generation
ICNOT	Options processor
ICNRP[1]	Report generation
ICNSM[1]	String manipulation
ICNST	Storage manager
ICNSY[1]	Symbol table
ICNTC[1]	Type conversion
ICNUS[1,2]	User generation utilities
ICNUT	Operating system utilities

[1] Not valid for TRDATA, NOTRDATA, TRGLOB, or NOTRGLOB. [2] Not applicable to EP standalone.

Other NDF modules, not falling under these categories, are statement and keyword routines specified by SK.

NTuneMON Keywords: The following diagrams show the NTuneMON keywords that can be coded on this definition statement. They are required if you have included the NTuneMON tuning feature, NTuneNCP, in your communication controller. See the *NTuneMON V2R5 User's Guide* for descriptions of these keywords.

►►──USERGEN=ATFTUNE──►◄

Keyword Descriptions
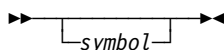
symbol

►►─┬──────────┬─►◄
   └─*symbol*─┘

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

ADDRCONV

```
►►──────┬─────────────────────────────┬──────►◄
        └─ADDRCONV=─┬─D──┬─┘
                    ├─H──┤
                    └─DD─┘
```
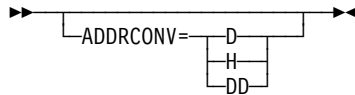
Default: None

Converts the ADDR keyword on frame relay PU statements to the DLCI keyword in a format of your choice (decimal, hexadecimal, or dotted decimal).

ADDRCONV can be coded on either the JCL parm list or the generation definition OPTIONS statement. If it is coded in both places, the OPTIONS specification overrides the JCL. NDF can:

* Process ADDRCONV independent of a migration aid run
* Write ADDRCONV to the NEWDEFN file

D   Indicates decimal.

H   Indicates hexidecimal.

DD   Indicates dotted decimal.

CHANNELS

```
            ┌─CHANNELS=GROUP─┐
►►──────────┼────────────────┼──►◄
            └─CHANNELS=GROUP─┘
```
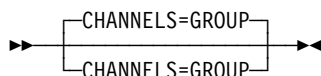
Use: Migration aid function
Default: GROUP

Specifies that 370 I/O channel attachments are to be defined on the GROUP, LINE, and PU definition statements in the output generation definition (NEWDEFN file).

You can code CHANNELS in your NDF job control language (JCL) or VM generation EXEC as well as on the OPTIONS definition statement. If you code CHANNELS in both places, the OPTIONS definition statement overrides the JCL or VM generation EXEC.

DPU

```
                  ┌─DPU=YES─┐
  ►►─────┬─────────────────┬──────►◄
         └─DPU=──┬─YES─┐────┘
                 └─NO──┘
```
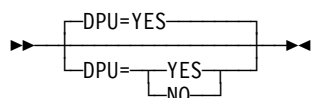
Use: Migration aid function
Default: YES

Specifies whether dynamic path update keywords are to be allowed to default, thereby adding dynamic path update capability to NCP. Dynamic path update is unlike most NCP functions in that the keyword defaults cause NCP storage to be allocated for the function. Code DPU=NO to cause the migration aid to set any unspecified dynamic path update keyword or suboperand to zero, thereby saving NCP storage.

You can code DPU in your NDF JCL or VM generation EXEC as well as on the OPTIONS definition statement. If you code DPU in both places, the OPTIONS definition statement overrides the JCL or VM generation EXEC.

FASTRUN

```
  ►►──────┬──────────────┬──────►◄
          └─FASTRUN=ON───┘
```

| Use: NCP or EP

Specifies whether the NDF generation process is to generate table and link-edit source. If you specify FASTRUN=ON, NDF checks the syntax of the input generation definition but does not generate table and link-edit source. If you code FASTRUN, the OPTIONS definition statement must be the first definition statement in the generation definition.

To use the FASTRUN option, use JCL or EXECs that call only NDF. See the *NCP, SSP, and EP Generation and Loading Guide* for sample JCL and EXECs.

NDNAME

```
  ►►──────┬──────────────────┬──────►◄
          └─NDNAME=name──────┘
```
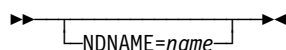
| Use: NCP or EP

Supports the automatic recataloging of the NEWDEFN file when NDF is running under VSE. *name* specifies a file name with a maximum length of 8 characters. The first character of the file name must be one of the following characters: A through Z, $, #, or @. The remaining characters must be chosen from the following: A through Z, 0 through 9, $, #, or @.

Note: To run a generation definition in which you specify NDNAME, you must change your JCL. See the *NCP, SSP, and EP Generation and Loading Guide* for more information.
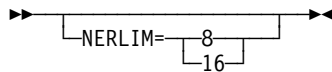
NERLIM

```
►►─┬──────────────────┬─►◄
   └─NERLIM=─┬──8──┬──┘
            └──16──┘
```

Use: Migration aid function
Default: None

Specifies that the migration aid is to add the ERLIMIT keyword to any NETWORK definition statement on which it is not coded, and specifies the value for it.

NEWDEFN

```
►►─┬──────────────────────────────────────────────────────────────────┬─►◄
   └─NEWDEFN=─┬─NO──────────────────────────────────────────────┬─┘
             │          ┌─ECHO─┐    ┌─NOSUPP─┐    ┌─REUSE───┐   │
             └─(YES,──┼──────┼──,──┼────────┼──,──┼─────────┼──)─┘
                      ├─ECHO─┤     └─SUPP───┘     ├─REUSE───┤
                      └─PACK─┘                    └─NOREUSE─┘
```
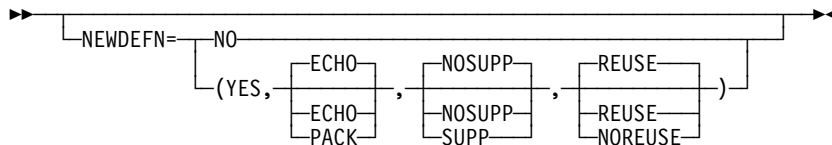
Example: NEWDEFN=(YES,,SUPP)

Specifies whether NDF is to create a new generation definition, or *NEWDEFN file*, during the generation process, and specifies certain characteristics of the NEWDEFN file. See the *NCP, SSP, and EP Resource Definition Guide* for a description of NEWDEFN files.

If you code the NEWDEFN keyword, the OPTIONS definition statement must be the first definition statement in the generation definition.

NO│YES

Default: See text

Specifies whether NDF is to create a NEWDEFN file. When you define any of the following resources, you must specify YES.

- NCP/Token-Ring interconnection (NTRI)
- 3746 Model 900 Token Ring
- Frame relay
- ISDN
- Ethernet-type LANs
- Internet Protocol
- Network Routing Facility (NRF)
- X.25 NCP Packet Switching Interface (NPSI)
- Automatic line generation using the AUTOLINE or AUTOCOPY keyword on the LINE definition statement
- User-written generation applications that modify the generation definition

When you run the migration aid function or code REMOVCOM=YES on the OPTIONS definition statement or in your NDF JCL or VM generation EXEC, the default is NEWDEFN=YES. Otherwise, the default is NEWDEFN=NO.

Migration Aid Function: When you run the migration aid function or code REMOVCOM=YES on the OPTIONS definition statement or in your NDF JCL or VM generation EXEC, if you code NEWDEFN=NO, NDF

overrides it with YES, but writes NO to the OPTIONS definition state-
ment in the NEWDEFN file.

ECHO│PACK

Default: ECHO

If you specify (YES,PACK), NDF reduces the size of the NEWDEFN file by
removing comments and coding as many keywords as possible on each line. If
you specify (YES,ECHO), NDF makes the NEWDEFN file resemble the input
generation definition as closely as possible, including comments.

Migration Aid Function: When you run the migration aid function or when
you code REMOVCOM=YES on the OPTIONS definition statement or in
your NDF JCL or VM generation EXEC, if you specify PACK, the
migration aid overrides it with ECHO, but writes PACK to the OPTIONS
definition statement in the NEWDEFN file.

NOSUPP│SUPP

Default: NOSUPP

If you specify NOSUPP, NDF adds comments to the NEWDEFN file to identify
the generation load module that passes the definition statements and
keywords. If you specify SUPP, these comments are suppressed.

REUSE│NOREUSE

Default: REUSE

If you specify REUSE, NDF builds the NEWDEFN file so that it can be reused
as the generation definition if you have Token-Ring, X.25 NPSI, or frame-relay
definition statements that are automatically generated. If you specify
NOREUSE, the NEWDEFN file is not built to be reused. See the *NCP, SSP,
and EP Resource Definition Guide* for more information.

Migration Aid Function: When you run the migration aid function or when
you code REMOVCOM=YES on the OPTIONS definition statement or in
your NDF JCL or VM generation EXEC, if you code NOREUSE, the
migration aid overrides it with REUSE, but writes NOREUSE to the
OPTIONS definition statement in the NEWDEFN file.

NOTRDATA
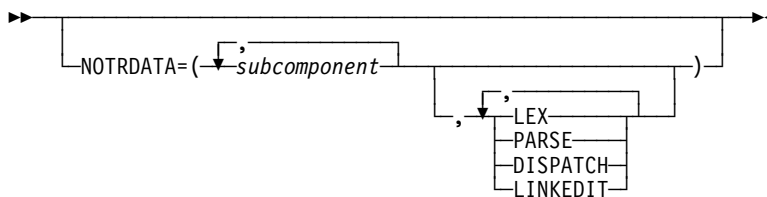
Use:	NCP or EP
Default:	None
Example:	NOTRDATA=(ICNIO,ICNIP,DISPATCH)

Specifies the subcomponent or data areas for which the data trace is deactivated. Specify NOTRDATA if your IBM support representative instructs you to run a data trace for diagnostic purposes.

subcomponent
> Specifies the name of a subcomponent for which the data trace is to be deactivated. For example, if you specify NOTRDATA=(ICNIO,ICNOB), the data trace is deactivated for all modules in the IO and OB subcomponents. See page 2-341 for a list of NDF subcomponents.

LEX
> Specifies that NDF is not to print each lexical token, length, and type as soon as it is known.

PARSE
> Specifies that NDF is not to print the parsed representation of each statement as soon as it is known.

DISPATCH
> Specifies that NDF is not to print an identifying message for each routine in the keyword vector table (KVT) that is called.

LINKEDIT
> Specifies that NDF is not to trace the generation of the linkage editor control cards.

NOTRGLOB

```
►►─┬──────────────────────────────┬─►◄
   │         ┌─;──────────┐        │
   └─NOTRGLOB=(─▼─global_trace─┴─)─┘
```
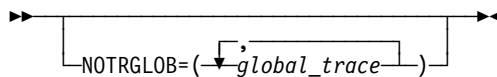
| Use: NCP or EP
Default: None

Deactivates global trace activity for the specified global traces.

NOTRPARM
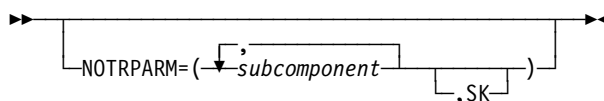
```
►►─┬──────────────────────────────────┬─►◄
   │         ┌─;──────────┐            │
   └─NOTRPARM=(─▼─subcomponent─┴─┬──────┬─)─┘
                                 └─,SK─┘
```

| Use: NCP or EP
Default: None

Specifies the subcomponent or statement and keyword routines for which the I/O parameter trace is deactivated. Specify NOTRPARM if your IBM support representative instructs you to run a parameter trace for diagnostic purposes.

subcomponent
> Specifies the name of a subcomponent for which the I/O parameter trace is to be deactivated. For example, if you specify NOTRPARM=(ICNSM,ICNSY), the I/O trace is deactivated for all modules in the SM and SY subcomponents. See page 2-341 for a list of NDF subcomponents.

SK

Specifies that NDF is not to trace I/O parameters for statement and keyword routines.

NOTRPROC
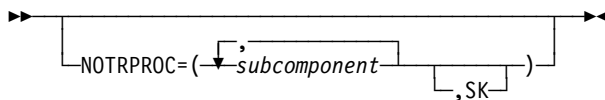
```
►►─┬─────────────────────────────────────────────────┬─►◄
   │            ┌──────,────────┐                     │
   └─NOTRPROC=(─▼─subcomponent──┴──┬──────────┬──)─┘
                                   └──,SK──┘
```

Use: NCP or EP
Default: None

Specifies the subcomponent or statement and keyword routines for which the procedure entry and exit trace is to be deactivated. Specify NOTRPROC if your IBM support representative instructs you to run a procedure trace for diagnostic purposes.

subcomponent

Specifies the name of a subcomponent for which the entry and exit trace is to be deactivated. For example, if you specify NOTRPROC=(ICNSM,ICNSY), the entry and exit trace is deactivated for all modules in the SM and SY subcomponents. See page 2-341 for a list of NDF subcomponents.

SK

Specifies that NDF is not to trace procedure entry and exit for statement and keyword routines.

REMOVCOM

```
          ┌─REMOVCOM=NO────┐
►►─┬───────┴────────────────┴─┬─►◄
   └─REMOVCOM=─┬──YES──┬────┘
              └──NO───┘
```
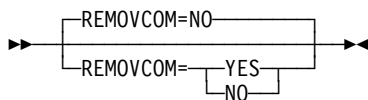
Use: Migration aid function
Default: NO

Specifies whether NDF is to remove the comments added to the NEWDEFN file by the migration aid function. REMOVCOM is meaningful only after you use the migration aid function.

You can code REMOVCOM in your NDF JCL or VM generation EXEC, as well as on the OPTIONS definition statement. If you code REMOVCOM in both places, the OPTIONS definition statement overrides the JCL or VM generation EXEC.

SAVEADDR
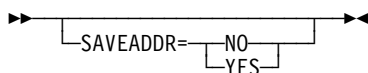
```
►►─┬─────────────────────────┬─►◄
   └─SAVEADDR=─┬──NO──┬────┘
              └──YES─┘
```

Use: Migration aid function
Default: See text

Specifies whether hardware-specific addresses coded for the ADDRESS, AUTO, DUALCOM, DYNADMP, HICHAN, and LOCHAN keywords on the BUILD and LINE

definition statements in the input generation definition are to be preserved in the output generation definition (NEWDEFN file). Code SAVEADDR=YES to preserve the addresses. Code SAVEADDR=NO if your configuration does not use the same addresses.

The default is SAVEADDR=YES when either of the following is true:

- TMODEL equals the value of MODEL on the BUILD definition statement.

- TMODEL=3745-210, 3745-21A, 3745-310, 3745-31A, 3745-410, 3745-41A, 3745-610, or 3745-61A, and MODEL=3745 on the BUILD definition statement.

Otherwise, the default is SAVEADDR=NO.

When migrating to a different communication controller model, some of the addresses specified by hardware-specific keywords may not be valid for the target controller. If you specify SAVEADDR=YES, the migration aid replaces most addresses not valid with a string of three question marks (???); in a few cases, the migration aid issues an error message indicating an address is not valid. You must manually change the values of these keywords in the NEWDEFN file to specify valid addresses.

If you specify SAVEADDR=NO, the migration aid replaces the values of all hardware-specific keywords with question marks regardless of whether the addresses are valid for the target controller.

You can code SAVEADDR in your NDF JCL or VM generation EXEC, as well as on the OPTIONS definition statement. If you code SAVEADDR in both places, the OPTIONS definition statement overrides the JCL or VM generation EXEC.

TMODEL

```
►►──┬──────────────────────────┬──►◄
    └─TMODEL=─┬─3745-130─┬──────┘
             ├─3745-150─┤
             ├─3745-160─┤
             ├─3745-170─┤
             ├─3745-17A─┤
             ├─3745-210─┤
             ├─3745-21A─┤
             ├─3745-310─┤
             ├─3745-31A─┤
             ├─3745-410─┤
             ├─3745-41A─┤
             ├─3745-610─┤
             └─3745-61A─┘
```
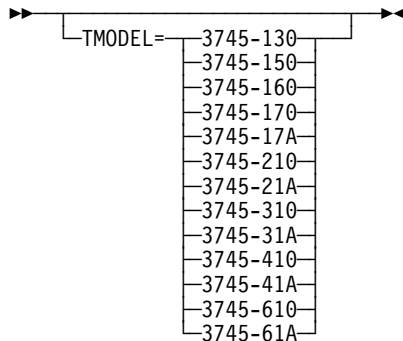
Use: Required for the migration aid function
Default: None

Specifies the communication controller model to which you are migrating.

You can code TMODEL in your NDF JCL or VM generation EXEC, as well as on the OPTIONS definition statement. If you code TMODEL in both places, the OPTIONS definition statement overrides the JCL or VM generation EXEC.

TOCONV

```
►►──┬─────────────────────────────┬──►◄
    └─TOCONV=─┬─YES─┬─────────────┘
             └─NO──┘
```
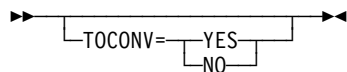
Default: None

Replaces the LOCALTO and REMOTTO keywords on Token-Ring and frame relay lines with the T1TIMER keyword.

TOCONV can be coded on either the JCL parm list or the generation definition OPTIONS statement. If it is coded in both places, the OPTIONS specification over-rides the JCL. NDF can:

• Process TOCONV independent of a migration aid run
• Write TOCONV to the NEWDEFN file

YES
   Indicates you want to replace LOCALTO and REMOTTO with T1TIMER

NO Indicates you do not want to replace LOCALTO and REMOTTO with T1TIMER

TRDATA

```
►►──┬────────────────────────────────────────────────────┬──►◄
    │             ┌─,─────┐                                │
    └─TRDATA=(─▼─subcomponent─┬──────────────────────┬──)─┘
                              │          ┌─,──────┐   │
                              └─,─▼─┬─LEX──────┬──┘
                                   ├─PARSE────┤
                                   ├─DISPATCH─┤
                                   └─LINKEDIT─┘
```
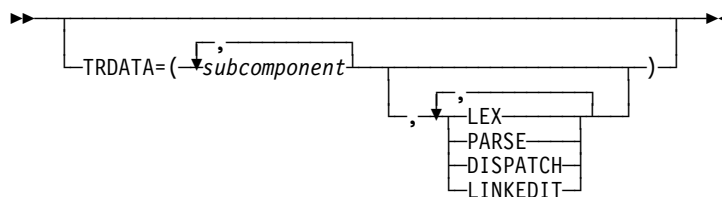
Use: NCP or EP
Default: None
Example: TRDATA=(ICNIO,ICNIP,DISPATCH)

Specifies the subcomponent or data areas for which the data trace is to be acti-vated. Code TRDATA if your IBM support representative instructs you to run a data trace for diagnostic purposes.

subcomponent
   Specifies the name of a subcomponent for which the data trace is activated. For example, if you specify TRDATA=(ICNIO,ICNOB), the data trace is acti-vated for all modules in the IO and OB subcomponents. See page 2-341 for a list of NDF subcomponents.

LEX
   Specifies that NDF is to print each lexical token, length, and type as soon as it is known.

PARSE
   Specifies that NDF is to print the parsed representation of each statement as soon as it is known.

DISPATCH
   Specifies that NDF is to print an identifying message for each routine in the keyword vector table (KVT) that is called.

LINKEDIT
Specifies that NDF is to trace the generation of the linkage editor control cards.

TRGLOB

```
►►─┬──────────────────────────────────────────┬─►◄
   │          ┌─────,─────┐                    │
   └─TRGLOB=─┬─(──▼─global_trace─┬─)──────────┬┘
            ├─HALT──────────────────┤
            └─RESUME────────────────┘
```
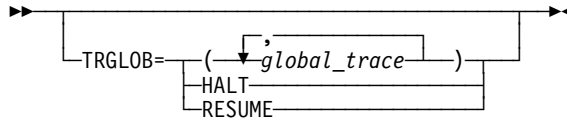
| Use: NCP or EP
Default: None

Specifies the global variables for which the global trace is to be activated. Code
TRGLOB if your IBM support representative instructs you to run a global trace for
diagnostic purposes.

You may activate a trace for a maximum of 24 global traces at one time on one or
more OPTIONS definition statements. TRGLOB must be coded in conjunction with
TRPROC to activate the trace activity.

Code TRGLOB=HALT or TRGLOB=RESUME to halt or resume the printing of trace
data at any time.

TRPARM
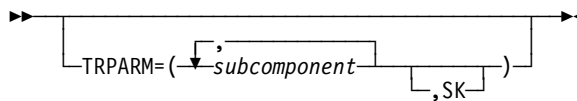
```
►►─┬───────────────────────────────────────┬─►◄
   │          ┌─────,─────┐                 │
   └─TRPARM=(──▼─subcomponent─┬──────┬─)────┘
                             └─,SK─┘
```

| Use: NCP or EP
Default: None

Specifies the subcomponent or statement and keyword routines for which the I/O
parameter trace is to be activated. Specify TRPARM if your IBM support represen-
tative instructs you to run a parameter trace for diagnostic purposes.

subcomponent
Specifies the name of a subcomponent for which the I/O parameter trace is
activated. For example, if you specify TRPARM=(ICNSM,ICNSY), the I/O trace
is activated for all modules in the SM and SY subcomponents. See page
2-341 for a list of NDF subcomponents.

SK
Specifies that NDF is to trace I/O suboperands for the statement and keyword
routines.

TRPROC
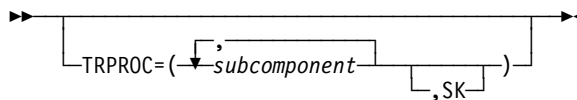
```
►►─┬───────────────────────────────────────┬─►◄
   │          ┌─────,─────┐                 │
   └─TRPROC=(──▼─subcomponent─┬──────┬─)────┘
                             └─,SK─┘
```

| Use: NCP or EP
Default: None

Specifies the subcomponent or statement and keyword routines for which the procedure entry and exit trace is to be activated. Specify TRPROC if your IBM support representative instructs you to run a procedure trace for diagnostic purposes.

subcomponent
> Specifies the name of a subcomponent for which the entry and exit trace is to be activated. For example, if you specify TRPROC=(ICNSM,ICNSY), the entry and exit trace is activated for all modules in the SM and SY subcomponents. See page 2-341 for a list of NDF subcomponents.

SK
> Specifies that NDF is to trace procedure entry and exit for statement and keyword routines.

TRSNAP
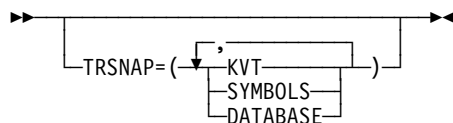
Use:	NCP or EP
Default:	None

Specifies which data structures NDF is to print in a readable format.

KVT
> Specifies that NDF is to print all KVTs.

SYMBOLS
> Specifies that NDF is to print the symbol table.

DATABASE
> Specifies that NDF is to print the storage manager control structures and data sets.

TUSGTIER
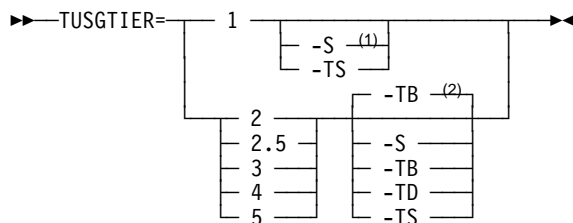
[1] For usage tier 1, the -S suffix is not valid for 3745-410/41A/610/61A.

[2] Defaults for usage tiers 2 through 5:

- -TB is the default for tiers 2 and 2.5, but only if MODEL=3745-41A or 3745-61A. Otherwise there is no default for tiers 2 and 2.5.

- -TB is the default for tier 3.

- -TB is not a default for usage tiers 4 and 5. There is no default for usage tiers 4 and 5.

Use:	Required for the migration aid function

Default: See the second note under the USGTIER syntax diagram.

Specifies the usage tier for the NCP version and communication controller to which you are migrating. Your system engineer can determine the usage tier from the HONE Configurator hardware configuration output report (CF37xx Report).

You can code TUSGTIER in your NDF JCL or VM generation EXEC, as well as on the OPTIONS definition statement. If you code TUSGTIER in both places, the OPTIONS definition statement overrides the JCL or EXEC.

EP Standalone: The TUSGTIER keyword is not valid.

See USGTIER on the BUILD definition statement on page 2-59 for more information.

TVERSION

```
►►──TVERSION=──┬──VxRy──┬──────►◄
              └──VxRyF──┘
```

Use: Required for the migration aid function
Default: None

Specify the NCP or EP version (Vx) and release (Ry) you are migrating to. You cannot migrate to a version and release that is earlier than the NCP or EP currently loaded. The version and release specified by TVERSION must be the same as or later than the version and release specified by the VERSION keyword on the BUILD definition statement.

You can code TVERSION in your NDF JCL or VM generation EXEC, as well as on the OPTIONS definition statement. If you code TVERSION in both places, the OPTIONS definition statement overrides the JCL or EXEC.

3746 Model 900 CSS:

1. To specify that you want NCP support for the 3746 Model 900 connectivity subsystem, use the VxRyF suboperand. (Note that VxRyF is valid only for V6R2 or later.)

2. Please note that you must obtain a license for the 3746 Model 900 CSS feature. Refer to Order instructions in the NCP announcement letter for ordering information.

EP Standalone: To migrate to EP R14 standalone, code TVERSION=V1R14.

USERGEN

```
►►──┬───────────────────────────┬──►◄
    │           ┌──;──┐          │
    └──USERGEN=(──▼─name─┴──)──┘
```

Default: None

Specifies the names of the generation application load modules that use the NDF standard attachment facility. Each user-written generation application must have a separate generation load module to process its statements and keywords and to generate table source statements. Define a maximum of 25 generation application load modules in a single generation. The first 3 characters of each generation application load module name must be unique and cannot be either ECL or ICN.

If you code USERGEN, the OPTIONS definition statement must be first in the generation definition.

Note: The start record in the SKVT of each generation load module indicates to NDF whether the load module name will be valid on the VIROWNER or LNKOWNER keywords. See the *NCP and SSP Customization Guide* for more information on user-written generation applications.

EP Standalone: The USERGEN keyword is not valid.

PATH Definition Statement

| ***Usage:*** Required. Not applicable to EP standalone. One for each destination subarea that NCP routes data to within a particular network. If you define a gateway NCP, these destination subareas include subareas in the native network and in the other networks that NCP joins.

The settings you select for transmission group flow-control thresholds and virtual route window sizes can significantly affect the performance of your network. The correct value for these settings is determined by your network's configuration and data flow. Use the Network Traffic Analyzer (NTA) to:

- Optimize your network data flows by identifying SNA traffic congestion problems
- Calculate correct virtual route parameter settings by recommending actions that tune virtual route performance

Position: Must precede the first GROUP definition statement. If defining a gateway NCP, and when defining routes to destination subareas in the native network, must follow the GWNAU definition statements that define cross-network session specifications for resources in another network that can participate in sessions with resources in the native network. When you are defining routes to destination subareas in a network other than the native network, PATH must follow GWNAU definition statements defining cross-network session specifications for resources in the native network that can participate in sessions with resources in the other network.

When you are defining PATH for a model network, the PATH definition statement must follow the GWNAU definition statement defining the model network. You must define at least one PATH statement for a line using a dynamic entry in the NVT to have a transmission group attached during link station activation. See COPIES on the NETWORK definition statement on page 2-332 for more information.

Keywords: Table 2-99 shows the keywords described under this definition statement.

Table 2-99. Keywords on the PATH Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
DESTSA	2-355	Yes			
ER*n*	2-355				Yes*
VR*n*	2-357				
VRPWS*nm*	2-358				Yes*

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►─┬─────────┬─►◄
   └─symbol──┘
```
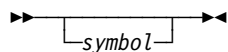
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

DESTSA

```
►►──DESTSA=(─┬─address─┬─)──►◄
            └────;────┘
```
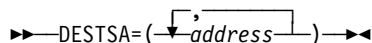
Use: Required
Default: None
Minimum: 1
Maximum: See text

Specifies the subarea address of one or more destination subareas, which can also be adjacent subareas, to which this NCP routes PIUs using the explicit routes defined by this PATH definition statement. The value must be numeric. A subarea specified by DESTSA must not appear in another DESTSA keyword within this network. Although you can specify more than one subarea address, it is recommended that you code only one subarea address with the DESTSA keyword and that you code one PATH definition statement for each destination subarea.

At least one subarea address is required. The maximum address is equal to the value of SALIMIT on the corresponding BUILD or NETWORK definition statement.

ERn

```
►►─┬──────────────────────────────────────┬─►◄
   └─ERn=(─┤ Explicit Route Parameters ├─)─┘
```

Explicit Route Parameters:

```
├─adjsa─,─┬─1────┬─,─┬─5000──────┬─,─┬─5000──────┬─,─┬─5000─────┬─,──────►
          └─tgn──┘   └─lothresh──┘   └─medthresh─┘   └─hithresh─┘

►─┬─20000──────┬─┤
  └─totthresh──┘
```
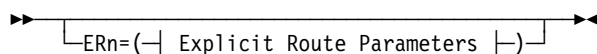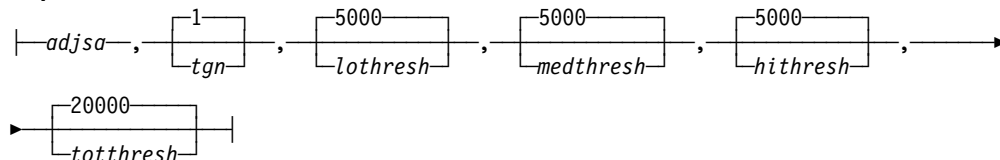
Example: ER1=(30,2,,,10000)

Specifies the adjacent subarea, transmission group number, and transmission group flow-control thresholds for explicit routes 0 to 15. In ER*n*, *n* can be 0 to the value of ERLIMIT minus 1.

A subarea can be both a destination subarea and an adjacent subarea. The minimum subarea address you can code is 1. The maximum address is the value of SALIMIT on the corresponding BUILD or NETWORK definition statement.

For a PATH definition statement on the native network, use the values coded for ERLIMIT on the BUILD definition statement to validate *n*.

For a PATH definition statement on non-native networks, use the values coded for ERLIMIT on the NETWORK definition statement for that network to validate *n*. If ERLIMIT is not coded on the NETWORK statement, use the value for the BUILD definition statement.

adjsa

> Default: None

> Specifies the address of an adjacent subarea (host or NCP) to receive PIUs on this explicit route bound for one of the destination subareas defined for DESTSA.

tgn

> Default: 1
> Minimum: 1
> Maximum: 255

> Specifies the transmission group number for this explicit route. The transmission group number is an arbitrary number that you assign to a transmission group for identification. Also use the value chosen for the *tgn* parameter for an explicit route in the adjacent subarea that communicates with this subarea. The same transmission group can be used for any or all of the remaining explicit routes, depending on the network configuration.

The following suboperands define flow-control thresholds for PIUs transmitted over this transmission group. Because the first ER*n* keyword for a given *adjsa,tgn* pair defines the transmission group, code these flow-control thresholds only on that keyword. Any values you code for these thresholds on a subsequent ER*n* keyword for the same *adjsa,tgn* pair are ignored.

lothresh
medthresh
hithresh

> Default: 5000
> Minimum: 1
> Maximum: 524288 (NCP V6R2 and earlier)
> 4194304 (NCP V6R3 and later)

> Define the flow-control thresholds for low-priority, medium-priority, and high-priority PIUs transmitted over this transmission group. When the total size of all the PIUs of a particular priority level queued for transmission over this transmission group exceeds the number of bytes you specify for the corresponding threshold, NCP informs the adjacent subarea that the transmission group is congested.

totthresh

> Default: 20000
> Minimum: 1
> Maximum: 2097152 (NCP V6R2 and earlier)
> 4194304 (NCP V6R3 and later)

Defines the flow-control threshold for all PIUs transmitted over this transmission group. When the total size of all the PIUs of all priority levels queued for transmission over this transmission group exceeds the number of bytes you specify for this threshold, NCP informs the subarea node or nodes that originated the queued data that the transmission group is congested. Those subarea nodes will lower their current VR transmit window sizes, which will reduce the amount of data being added to the queue.

This value also defines the limit for the number of bytes of APPN HPR data that can be queued on this transmission group at any time. If the number of bytes of APPN HPR data queued exceeds the number of bytes you specify for this threshold, NCP discards additional APPN HPR data that arrives instead of queueing it. This causes the RTP endpoints to reduce the rate at which they send APPN HPR data into the subarea network.

VRn

```
►►─────────────────────►◄
   └─VRn=ernumber─┘
```
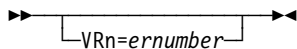
Default: See text

Specifies the virtual route to explicit route mapping that NCP uses when activating virtual routes. In VR*n*, *n* can be 0 to 7. *ernumber* must be the same as the number of the explicit route to which you are assigning this virtual route.

Code these keywords (or a subset of them as needed) if you want NCP to be able to activate this virtual route, or if this NCP is a gateway NCP. The default is a null mapping.

On a PATH definition statement for the native network, *ernumber* must be less than the value of ERLIMIT on the BUILD definition statement.

On a PATH definition statement for a non native network, *ernumber* must be less than the value of ERLIMIT on the NETWORK definition statement for the non-native network. If ERLIMIT is not coded on the NETWORK definition statement, *ernum* must be less than the value of ERLIMIT on the BUILD definition statement.
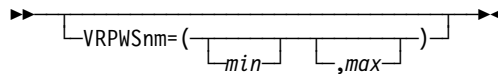
VRPWSnm

```
►►──┬─────────────────────────────────────┬──►◄
    └─VRPWSnm=(──┬──────┬──┬────────┬──)─┘
                 └─min─┘  └─,max─┘
```

Default: See Text
Minimum: 1
Maximum: 255

Specifies the minimum and maximum virtual route pacing window sizes to support high-speed or long-delay transmission facilities for virtual routes.

The last 2 characters of this keyword identify the virtual route whose pacing window sizes you are defining, as follows:

n

Specifies the virtual route number (defined by VR*n* on this PATH definition statement). Valid values are 0 to 7.

m

Specifies the priority of the virtual route. Valid values are 0, 1, and 2.

When the transmission group (TG) is a channel (CA=TYPE6, CA=TYPE7, or ESCON) and the destination subarea is adjacent to the NCP, VRPWSnm defaults to:

- A minimum window size of 1
- A maximum window size of 2 times MAXBFRU (2 * MAXBFRU) not to exceed 254

When the destination subarea is not adjacent to the NCP, VRPWSnm defaults to:

- A minimum window size equal to the hop count
- A maximum window size equal to three times the hop count

PCCU Definition Statement

Usage: VTAM. One for each VTAM that can activate NCP. This definition statement provides information to the VTAM initialization process. It is not required in the generation definition you use to generate NCP.

Position: Must precede BUILD definition statement.

Keywords: See the *VTAM Resource Definition Reference* for the description of this definition statement and its keywords.

PU Definition Statement

Usage: One PU definition statement for each device and link station.

Channel Links: The PU definition statement is not valid if you code LNCTL=CA and NCPCA=INACTIVE, or if you code LEVEL5=USER on the GROUP definition statement. Only one PU definition statement is allowed if you code LNCTL=CA and NCPCA=ACTIVE on the GROUP definition statement.

SDLC Links: You can define only one physical unit for each secondary, non-switched SDLC line.

Ethernet-Type LANs: You can define only one physical unit for each connection to an Ethernet-type LAN.

NTRI Frame Handler Logical Lines: You can code PU definition statements immediately following the LINE definition statement for a NTRI frame handler logical line. You can define multiple frame-relay physical units. The first PU defined is the link station subport. Additional PUs are NTRI logical FHSPs.

Internal NCP to Model 900 Point-to-Point IP Connection: Code one PU definition statement with PUTYPE=1 AND NETWORK=IP beneath a LINE statement that defines a 3746 Model 900-attached Token-Ring physical link.

Frame Relay: You can code frame-relay PU definition statements either immediately following the LINE definition statement for a frame-relay physical line or within a frame-relay logical group associated with a physical line by PHYSRSC on the GROUP definition statement. You can define more than one physical unit for each frame-relay nonswitched physical line. The first physical unit is the link station subport physical unit. Additional physical units are frame-relay frame handler physical units or frame-relay internet hosts. You can define only one physical unit for each frame-relay logical line and each frame-relay switched physical line. The link station subport physical unit station is the only physical unit allowed on a switched frame-relay physical line.

Position: Must follow the SERVICE definition statement or the LU definition statement for another device. If defining a link station or if no SERVICE definition statement is coded, must follow the LINE definition statement.

Keywords: Table 2-100 shows the keywords described under this definition statement.

Table 2-100 (Page 1 of 4). Keywords on the PU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ADDR	2-366				Yes*
ANS	2-369		G		

Sift:

B   BUILD definition statement
G   GROUP definition statement
L   LINE definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-100 (Page 2 of 4). Keywords on the PU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ARPTAB	2-371			(1000,20,NOTCANON)	Yes*
AVGPB	2-372		G		
BLOCK	2-373				Yes*
BNNSUP	2-374		G,L		
BRKCON	2-374		G,L	NONE	
CASDL	2-374		G,L		
COMRATE	2-375		G,L		Yes*
DATMODE	2-375		G,L		
DELAY	2-376		G,L	0.1	
DLCI	2-377				
DYNWIND	2-378			4,1,1,1,YES	Yes*
ERRORT	2-380		G,L	(3,4)	
FRSPRI	2-381		G,L	0	
GP3174	2-381		G,L	NO	
HPRQLIM	2-382		B,G,L	0	
INNPORT	2-382				
INTFACE	2-383				
IPQLIM	2-384				
IRETRY	2-384		G,L	NO	
ISDNCFG	2-385			CRC	
ISDNDSCT	2-385			15	
LADDR	2-385				
LANTYPE	2-386		G,L		
LLERP	2-387		G,L	NO	
LMI	2-388		G,L	CCITT	Yes*
LMODADR	2-389				
LOCALTO	2-389				
LPDA	2-390				
MAXBFRU	2-391		G,L	32	
MAXDATA	2-391		G,L		Yes*
MAXLU	2-394		G,L		
MAXOUT	2-395		G,L		Yes*
METRIC	2-399			1	

Sift:

B   BUILD definition statement
G   GROUP definition statement
L   LINE definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-100 (Page 3 of 4). Keywords on the PU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
MODULO	2-399		G,L	8	Yes*
MONLINK	2-400		G,L		
NETID	2-400		L		
NETWORK	2-401			SNA	
NEXT	2-401				
NPACOLL	2-403		G		
NPATP	2-404		G	NO	
PASSLIM	2-404		G,L		Yes*
PIUCHECK	2-408			NO	Yes*
PROTOCOL	2-408			RIP	
PUCB	2-408		G,L		
PUDR	2-408		G,L		
PUFVT	2-410		G,L		
PUNTFY	2-410		G,L		
PUTYPE	2-411		L		
P2PDEST	2-412				
REMOTTO	2-413		G		
RETRIES	2-414		G		Yes*
ROLE	2-414		G,L	NEG	
SEGMENT	2-415				
SNETMASK	2-415				
SPOLL	2-416		G,L	6	
SRT	2-416		G,L	(32768,32768,YES)	
SSAP	2-418			4	Yes*
TGCONF	2-418		G,L	MULTI	Yes*
TGN	2-419		G,L	1	
TIMEOUT	2-420				
TIMERS	2-421		G,L	(10,15)	
TRANSFR	2-422		B,G,L		
T1TIMER	2-423				
T2TIMER	2-424				
UNIQUE	2-426			YES	Yes*
XID	2-426		G,L		

Sift:

B   BUILD definition statement
G   GROUP definition statement
L   LINE definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
NTuneMON V2R5 User's Guide for more information.

Table 2-100 (Page 4 of 4). Keywords on the PU Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
XMONLNK	2-427		G,L		

Sift:

B   BUILD definition statement
G   GROUP definition statement
L   LINE definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements:
You can define most keywords of the LU definition statement on the PU definition statement. Additionally, you can code most PU keywords on the LINE or GROUP definition statement. Table 1-3 on page 1-12 shows which of the lower-level keywords you can code at a higher level.

If you specify NPARSC=YES, all keywords specified on this PU definition statement are ignored.

VTAM Keywords: The following diagrams show the VTAM keywords that can be coded on this definition statement. These keywords provide information to the VTAM initialization process. They are not required in the generation definition you use to generate NCP. See the *VTAM Resource Definition Reference* for descriptions of these keywords and the *VTAM Network Implementation Guide* for information on the VTAM initialization process.

VTAM keywords not listed below can be specified by coding VTM. as a prefix in front of the VTAM keyword.

```
►►─┬──────────────────────────────────────────────┬─►◄
    └─ASLENT=associated_LU_table_entry_name─┘
```

```
►►─┬──────────────────────────────────────┬─►◄
    └─ASLTAB=associated_LU_table_name─┘
```

```
►►─┬──────────────┬─►◄
    └─CPNAME=name─┘
```

```
       ┌─DISCNT=NO─────────────────────────────┐
►►─┼───────────────────────────────────────┼─►◄
       └─DISCNT=─┬─(─YES─┬─────────┬─)─┬──┘
                 │        ├─,F──────┤   │
                 │        └─,─┬─F──┬┘   │
                 │            └─NF─┘    │
                 └─NO───────────────────┘
```

```
►►─┬──────────────────────────────────────────────┬─►◄
    └─DLOGMOD=default_logon_mode_entry_name─┘
```

```
►►─┬─DYNLU=─┬─YES─┬─┬─►◄
    └        └─NO──┘ ┘
```

```
                         ┌─ENCR=NONE────────┐
 ►►──────────────────────┴──────────────────┴──────────────►◄
       └─ENCR⁽¹⁾=─┬─NONE─┬─
                  ├─REQD─┤
                  ├─SEL──┤
                  └─OPT──┘
```

¹ Valid for MVS only

```
                         ┌────────────────┐
 ►►──────────────────────┴────────────────┴──────────────────►◄
       └─FEATUR2=(─▼─keyword─┴─)─┘
```

```
 ►►───────────────────────►◄
       └─HPR=─┬─YES─┬─┘
              └─NO──┘
```

```
 ►►──────────────────────────►◄
       └─INCLUD0E=─┬─NO──┬─┘
                   └─YES─┘
```

```
 ►►──────────────────────────►◄
       └─INNPORT⁽¹⁾=─┬─NO──┬─┘
                     └─YES─┘
```
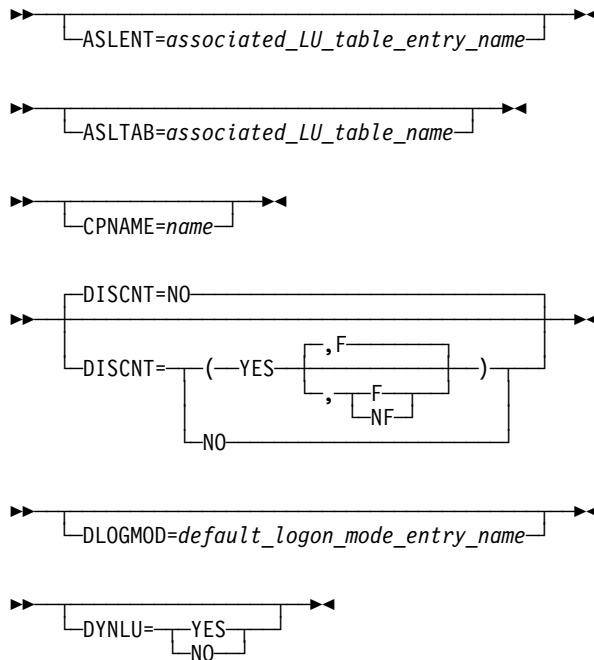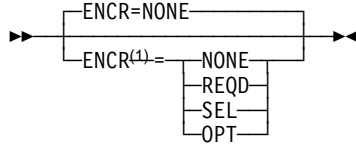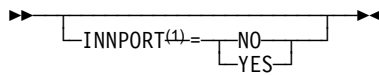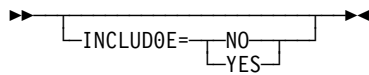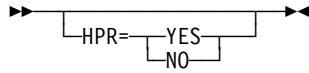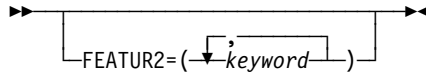
¹ Because of its importance to NCP, INNPORT is also described in this book.

```
       ┌─ISTATUS=ACTIVE──────────┐
 ►►────┴─────────────────────────┴──►◄
       └─ISTATUS=─┬─ACTIVE───┬─┘
                  └─INACTIVE─┘
```

```
       ┌─LIMRES=NO──────┐
 ►►────┴────────────────┴──►◄
       └─LIMRES=─┬─NO──┬─┘
                 └─YES─┘
```

```
 ►►──────────────────────────────────────────►◄
       └─LOGAPPL=application_program_name─┘
```

```
 ►►────────────────────────────────────►◄
       └─LOGTAB=interpret_table_name─┘
```

```
 ►►──────────────────────────────────────────►◄
       └─LUGROUP=logical_unit_group_name─┘
```

```
 ►►────────────────────────────────────────────►◄
       └─LUSEED=logical_unit_pattern_name─┘
```

```
 ►►──────────────────────────────────────────────►◄
       └─MDLENT=model_name_table_entry_name─┘
```

```
 ►►───────────────────────────────────────►◄
       └─MDLTAB=model_name_table_name─┘
```

```
 ►►──────────────────────────────────────►◄
       └─MODETAB=logon_mode_table_name─┘
```

```
►►──────┬─NATIVE=─┬─NO──┬────────►◄
                  └─YES─┘
```

```
►►──────┬─NETID=─┬─(network_id)─┬────►◄
                 └─(,NOXNETLS)──┘
```

```
►►──────┬─OWNER=owner_name─┬────►◄
```

```
►►──────┬─REGISTER=─┬─NO──────┬────►◄
                    ├─NETSRVR─┤
                    └─DIRSRVR─┘
```

```
        ┌─SECNET=NO──────────┐
►►──────┼────────────────────┼────►◄
        └─SECNET⁽¹⁾=─┬─NO──┬──┘
                     └─YES─┘
```

¹ Valid for VTAM Version 3 only

```
►►──────┬─SPAN=span_name─┬────►◄
```

```
        ┌─SSCPFM=FSS────────┐
►►──────┼───────────────────┼────►◄
        └─SSCPFM=─┬─FSS─────┬─┘
                  ├─USSSCS──┤
                  ├─USS3270─┤
                  ├─USS3275─┤
                  ├─USS3780─┤
                  └─USSNTO──┘
```

```
►►──────┬─SUBAREA=n─┬────►◄
```

```
►►──────┬─TERM=type─┬────►◄
```

```
►►──────┬─USSTAB=USS_definition_table_name─┬────►◄
```

```
►►──VTAMTOPO=─┬─NOREPORT─┬────►◄
              └─REPORT───┘
```

Keyword Descriptions

symbol

```
►►──symbol──►◄
```

Use: Required when you code the PU definition statement
Default: None

Provides a name for the physical unit. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates. If AUTOLINE is specified, you must embed the first suboperand of the ADDRESS keyword on the LINE definition statement in the symbol.

ADDR

```
►►─┬─────────────────┬─►◄
   └─ADDR=address─┘
```
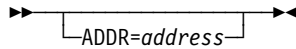
Use: Nonswitched links
Default: None

Specifies the hexadecimal representation of the 8-bit address of the physical unit represented by this PU definition statement. Any bit configuration except X'00' or X'FF' is valid. ADDR is not valid if you code DIAL=YES on the GROUP definition statement.

ADDR is required if the communication facility is nonswitched (DIAL=NO) and the physical unit is type 1 or 2.

ADDR is required if you code MODE=PRIMARY and do not code SDLCST on LINE for a primary multipoint subarea link.

ADDR is not valid if the connection is casual and you code MODE=SECONDARY on the GROUP definition statement.

| ADDR is not valid for a point-to-point (where SDLCST is coded on the LINE state-
| ment) subarea connection. In this configuration, NDF and NCP administer the
| appropriate address.

Do not code ADDR=FD for this physical unit if you have coded LPDATS=LPDA2 on the LINE definition statement.

NTRI Physical Lines: ADDR specifies the address of the NTRI link station.

For a NTRI link station supporting SNA logical lines (NETWORK=SNA on the PU definition statement), ADDR=01 is the only valid value.

For a NTRI link station supporting internet routing (NETWORK=IP), ADDR=02 is the only valid value.

For a NTRI link station supporting frame-relay traffic (NETWORK=FRELAY), ADDR=03 is the only valid value.

Token-Ring Logical Lines: ADDR is required for Token-Ring logical subarea lines. The ADDR value consists of a 2-character destination service access point (DSAP) followed by a 12-character medium access control (MAC) address. The valid format for ADDR is *aabccccccccccc*, with the following restrictions:

- *aa* (the DSAP) must be a nonzero hexadecimal multiple of 4.
- *b* must be 0–7, or 4–7 when connecting to another NCP.
- Each *c* must be X'0'–X'F'.

When you specify a DSAP other than 04, you must ensure that the destination node supports SSAPs other than 04. For instance, NCP V7R4 and earlier releases support only SSAP 04 for Token-Ring logical subarea connections.

When you specify a duplicate locally administered MAC address, **you must code UNIQUE=NO**. Only NCP V7R5 and later releases support duplicate locally administered MAC addressing for Token-Ring logical subarea connections. See UNIQUE on the PU definition statement on page 2-426 for additional information about destination MAC address uniqueness.

Do not specify X'aa000000000000' for the MAC address because it represents a null address. The MAC address portion cannot be the same as the LOCADD value for the physical connection of the logical line in this NCP; it must be the LOCADD value identifying the locally administered address of the physical Token-Ring connection in the adjacent NCP. This value is coded on the LINE definition statement in the adjacent NCP generation definition. For example, if LOCADD=400037450201 for the physical connection in the adjacent NCP, code ADDR=04400037450201 for this NCP. See LOCADD on the LINE definition statement on page 2-238 for additional information about MAC addresses of physical NCP Token-Ring connections.

ADDR is not valid for Token-Ring logical peripheral resources.

NTRI Frame Handler Logical Lines: ADDR must be specified for the first PU (link-station) on a NTRI frame handler logical line. In this case, ADDR is not a DLCI, but the DSAP and MAC address of the adjacent NCP. This is coded like a Token-Ring subarea logical line except that the DSAP must be X'C4'.

ADDR or DLCI must be coded for any additional PU (frame handler subports) to code the DLCI with the following rules:

- All DLCIs must be unique on the line.

- The maximum that can be specified with ADDR is X'FE'. (Use DLCI to code larger values.)

- The MAXFRAME limitation in the following frame relay section applies also.

3746 Model 900 Token-Ring Physical Lines: ADDR specifies the address of the link station.

For a link station supporting SNA logical lines (NETWORK=SNA on the PU definition statement), ADDR=01 is the only valid value.

For a link station supporting internet routing (NETWORK=IP), ADDR=02 is the only valid value.

Frame Relay: ADDR specifies the DLCI of the frame-relay subport.

Frame-Relay Internet Connection: Either ADDR or DLCI specifies the DLCI of the frame-relay internet host.

You can code either ADDR *or* DLCI, but not both on a PU definition statement to specify the address. The minimum value is X'10'.

Note: You can define more frame-relay resources if you code DLCI. For additional information on DLCI, refer to DLCI on the PU definition statement on page 2-377.

All DLCI values associated with a single frame-relay physical line must be unique. This includes those values defined outside NCP. For example, if you use DLCI X'30' on a frame-relay physical line to identify a subarea TESP, a frame handler subport (FHSP), or a frame-relay internet host, address X'30' cannot be used by a peripheral device to establish an incoming logical peripheral connection to NCP over that same physical line.

You should avoid using DLCI X'20' (decimal 32) when you define frame-relay peripheral links, since peripheral devices often use DLCI X'20' for incoming calls. When a subarea link TESP or FHSP uses this DLCI, the peripheral devices must be reconfigured to use a different DLCI.

Although DLCI values are coded in hexadecimal in an NCP generation definition, other number bases may be used for DLCI values defined by other products. For example, default DLCI values defined by the Integrated Digital Network Exchange (IDNX) are decimal numbers. If the LMI protocol is not supported for this line (LMI=NO on the first PU definition statement for this line), the maximum value for ADDR is the largest DLCI allowed by the line adapter for this line. For a TSS, HPTSS, or CLP adapter, the maximum is X'FE'.

If an LMI protocol is supported for this line (LMI is coded and is not equal to NO on the first PU definition statement for this line), the maximum value for ADDR is the lower of the following two values:

- The largest DLCI allowed by the line adapter for this line (see preceding text)

- The value of the following expression (converted to hexadecimal):

    $$((maxframe - fixed) / 5) + 15$$

    where:

    maxframe is the value of MAXFRAME for this line.

    fixed is 17 when LMI=CCITT or LMI=CCITT-NUI on the LMI PU definition statement.

    fixed is 18 when LMI=ANSI or LMI=ANSI-NUI on the LMI PU definition statement.

See the *NCP, SSP, and EP Resource Definition Guide* for more information about defining frame-relay subarea links.

Frame-Relay Physical Lines: For frame-relay physical lines, ADDR is not valid on the first PU definition statement for this line (the link-station subport) and is required on all subsequent PU definition statements (frame-relay frame handler subports and frame-relay internet connections).

Frame-Relay Logical Peripheral Lines: ADDR is not valid for frame-relay logical peripheral lines.

Boundary Access Node (BAN) for 3746 Model 900 Frame-Relay Logical Links: ADDR is required for 3746 Model 900 frame-relay line connections to remote Token Ring capable type 4 physical units through remote BAN

routers such as the IBM 6611 or IBM 2210. The ADDR value consists of a 2-character destination service access point (DSAP) followed by a 12-character medium access control (MAC) address. The valid format for ADDR is *aabccccccccccc*, with the following restrictions:

- The value of *aa* must be a nonzero hexadecimal multiple of 4.

- The value of *b* must be 0–7, or 4–7 when connecting to another NCP.

- Each *c* must be X'0'–X'F'.

Do not specify X'aa000000000000' for the MAC address because it represents a null address. The ADDR value does not have to be unique from other ADDR values on the associated physical line. The DLCI keyword is required to specify the DLCI for the subarea connection.

ESCON Links: ADDR is required for the logical stations defined for an ESCON logical link. The value that you code for the ADDR keyword must be the same as the value specified for the IODEVICE UNITADD keyword in the host operating system IOCDS definition. The valid range for ADDR is X'01' to X'10'. All ADDR values for a single logical line must be different.

ADDR is not valid for ESCON physical stations.

ANS
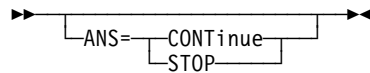
```
►►──┬─────────────────────────┬──►◄
    └─ANS=─┬─CONTinue─┬────────┘
           └─STOP─────┘
```

Default: See text

Specifies whether this device is to stop or continue to operate when NCP enters ANS.

If you specify ANS=STOP (valid only for type 1 and 2 physical units), NCP stops all communication with the device upon entering automatic shutdown. When the device detects that it can no longer communicate with NCP, it goes offline, regardless of whether it is currently participating in sessions.

ANS=STOP is the only valid choice for a device on a switched SDLC link if that link is not defined as user line control (that is, if LEVEL2 or VIRTUAL=YES is coded on the GROUP definition statement).

If you specify ANS=CONTINUE, NCP continues to service the link on which the device is located as long as the link and device remain operational. NCP does this even when it is in shutdown mode with respect to the access method with which it can no longer communicate. The device can continue to participate in sessions with access methods and application programs other than those affected by the shutdown.

If this PU definition statement represents a communication controller (that is, a type 4 physical unit), only ANS=CONTINUE is valid. CONTINUE is the default. For type 1 and type 2 physical units, ANS=STOP is the default if you omit this keyword.

If this physical unit is to participate in extended recovery facility (XRF) sessions, code ANS=CONTINUE.

ANS=CONTINUE is ignored by switched physical units, except for physical units attached with user line control to switched links, or physical units associated with Token-Ring lines.

ANS is not valid if you specify PUTYPE=4 and DIAL=YES for switched subarea support lines.

Attention: If you specify ANS=CONTINUE for a device on a switched link, NCP does not automatically deactivate the link, even if no sessions are active. You must use forced deactivation to disable the link unless the link is a switched subarea link and the link activity timer is running (BRKCON=CONNECTO or BRKCON=NOWNERTO). An attempt to reactivate the link without first using forced deactivation results in a negative response because the link is still active. Also, because the link is not deactivated automatically by NCP and thus can stay active for an indefinite period of time, the cost of the switched-link connection is likely to increase. Unless you have a switched subarea link, a link will be disabled when the link activity timer expires. If a backup communication management configuration (CMC) exists, you can use the VTAM switched major node DISCNT keyword to indicate that VTAM should deactivate a peripheral link after the last logical unit ends its session.

If you are using VTAM V3R1, session continuation for switched connections is supported only when LNCTL=USER or when the lines are NTRI logical lines.

If you are using VTAM V3R1.2 or a later VTAM release, session continuation for switched connections is fully supported. The ANS setting is always sent to NCP at activation time.

For subarea channel-link activation, code ANS when LNCTL=CA on the GROUP definition statement and PUTYPE=5. Only ANS=CONTINUE is valid. The default is CONTINUE if you code XMONLNK=YES.

Token Ring: If there is an owner of the physical line during automatic network shutdown, the line remains active when ANS=STOP or ANS=CONTINUE.

If the last owner of the line is no longer present, NCP checks the value of ANS coded on the PU definition statement:

- If ANS=STOP, the physical line is deactivated and the TIC is deinserted from the ring

- If ANS=CONTINUE is specified or is used by default, the physical line is not deactivated during automatic network shutdown, and the TIC remains inserted in the ring

If you do not code ANS and this PU definition statement defines a type 1 physical unit, NDF adds ANS=CONTINUE to the generation definition.

Code the same value for ANS on all PU definition statements under the LINE definition statement for a Token-Ring physical line.

NTRI Frame Handler Logical Lines: ANS=CONTINUE is the only valid value.

ISDN: ANS=CONTINUE is the only valid value for ISDN physical lines. If you do not code ANS, NDF adds ANS=CONTINUE to the generation definition.

For ISDN B-channels, code ANS=CONTINUE on the PU statement in the VTAM Switched Major Node definition deck. Because ISDN B-channels are switched links, NCP will not automatically deactivate them. Use the NCP ISDNDSCT keyword in the ISDN physical station definition to indicate that the 3746 Model 900 should deactivate a B-channel after the last associated logical station disconnects.

ESCON Links: For ESCON logical link stations defined with PUTYPE=2, you may code either ANS=CONTINUE or ANS=STOP. The default is ANS=STOP.

For ESCON logical link stations defined with PUTYPE=5 or for ESCON physical link stations, you may code only ANS=CONTINUE. The default is ANS=CONTINUE.

Table 2-101 shows the effect on ANS when you code certain keywords on the GROUP definition statement.

Table 2-101. NDF Values for ANS on the PU Definition Statement

When You Code This on GROUP	The Effect on ANS Is
ETHERNET=PHYSICAL	The default is ANS=CONTINUE. The only valid value is ANS=CONTINUE.
ECLTYPE=(LOGICAL,FRELAY)	NDF adds ANS=CONTINUE. The only valid value is ANS=CONTINUE.
ECLTYPE=PHYSICAL	NDF adds ANS=CONTINUE.
FRELAY=PHYSICAL	NDF adds ANS=CONTINUE. The only valid value is ANS=CONTINUE.
FRELAY=(LOGICAL,IP)	The default is ANS=CONTINUE. The only valid value is ANS=CONTINUE.
FRELAY=(LOGICAL,SUBAREA)	The default is ANS=CONTINUE. The only valid value is ANS=CONTINUE.
FRELAY=(LOGICAL,PERIPHERAL)	The default is ANS=STOP.
ISDN=PHYSICAL	The only valid value is ANS=CONTINUE.

ARPTAB
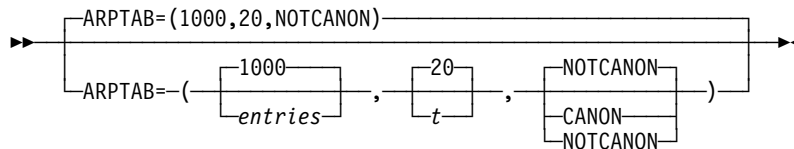
Use: Internet Protocol

Specifies the size of the ARP table and how NCP manages the entries in the table.

Token Ring: You can code ARPTAB for Token-Ring physical units only when you code NETWORK=IP on the PU definition statement.

Frame-Relay IP connections: You can code ARPTAB only on the LINE definition statement.

entries

Default: See text

Minimum: 1
Maximum: 65535

Specifies the maximum number of temporary entries that can exist in the ARP table at any given time. To optimize system operation, code a number equal to the maximum number of devices that you expect to be connected to the LAN.

Defaults to 4 for IP channel connections and for internal NCP to Model 900 point-to-point IP connections; otherwise 1000.

t

Default: 20
Minimum: 3
Maximum: 20

Specifies, in minutes, the transmit inactivity timer interval. This timer is used to reclaim temporary ARP table entries. If *t* minutes have elapsed since the last ARP exchange to the host associated with the entry, the entry is reclaimed and another ARP request must be sent to the host before more data can be trans-mitted to it.

This suboperand is not valid for IP channel connections or for internal NCP to Model 900 point-to-point IP connections.

NOTCANON | CANON

Use: Internet routing over NTRI

Specifies whether hardware addresses received in ARP packets are in canonical form. When ARPTAB=CANON, the hardware address in an ARP packet is interpreted using canonical, or inverted, format. In this format the *left*most byte of the 6-byte hardware address is transmitted first, and the *right*most bit of each byte is transmitted first.

When you specify ARPTAB=NOTCANON, the hardware address in an ARP packet is interpreted using noninverted format. In this format the *left*most byte of the 6-byte hardware address is transmitted first, and the *left*most bit of each byte is transmitted first.

Note: All Token-Ring internet devices that can receive or send ARP requests and responses to each other must use the same format.

AVGPB
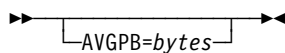
```
►►──┬────────────────┬──►◄
    └─AVGPB=bytes─┘
```

Default: None
Minimum: See text
Maximum: See text

Specifies the number of bytes used to control NCP buffer space for polled SDLC devices.

The minimum value for *n* is the value of BFRS on the BUILD definition statement. The maximum value is BRFS times 255. If you omit AVGPB, the default is seven times the value specified for BFRS on the BUILD definition statement.

If you define a type 2.1 physical unit, *n* is the maximum number of bytes expected from an SDLC device when it is polled for peripheral host node SDLC devices. For subarea host nodes, the value you use for AVGPB is the maximum number of bytes expected during a single data transfer operation. For better performance on subarea host node lines, be sure the value of AVGPB equals the number of bytes specified for TRANSFR on either the BUILD or LINE definition statement.

3746 Model 900 SDLC Lines: You can omit AVGPB since the 3746 Model 900 performs NCP data link control functions. If you code AVGPB, it will be ignored.

BLOCK
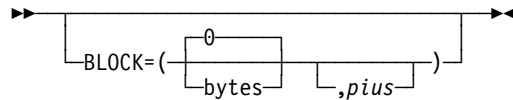
Specifies the maximum size, in bytes, for frames transmitted over this subarea link and the maximum number of PIUs for multiple-PIU frames.

bytes

> Default: 0
> Minimum: 0
> Maximum: 8192

> Specifies the maximum frame size in bytes. This value must be less than or equal to the value of MAXDATA on this PU definition statement.

pius

> Default: See text
> Minimum: 2
> Maximum: 16

> Specifies the maximum number of PIUs for each frame. The default is 0 if the value of *bytes* is 0; otherwise, the default is 8.

> **Attention:** If you have a PEP configuration and are using PIU blocking, you may need to reduce the maximum number of PIUs for each frame if an I/O interruption disrupts system operation or if there are overruns or underruns on EP lines.

BLOCK is not valid when you code the LEVEL2 keyword on the GROUP definition statement unless this physical unit is for a NTRI or frame-relay line.

Token Ring: BLOCK is valid only for logical subarea lines.

Frame Relay: BLOCK is valid only for frame-relay logical subarea lines.

ISDN: BLOCK is valid only for ISDN logical subarea lines.

User-Written Line Control: BLOCK is not valid for user-written line control.

BNNSUP

```
►►──────────────────────►◄
     └─BNNSUP=3270─┘
```
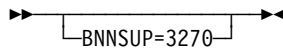
Use: Type 1 physical unit on a nonswitched link
Default: None

Specifies that the physical unit represented by this PU definition statement is a type 1 physical unit (PUTYPE=1) IBM 3270 (operating in SDLC mode.)

BNNSUP is not valid if the line is switched (DIAL=YES on the GROUP definition statement) or if the IBM 3270 is a type 2 physical unit (PUTYPE=2 on this definition statement).

BRKCON

```
            ┌─BRKCON=NONE────────┐
►►──┼────────────────────┼──►◄
     └─BRKCON=─┬─CONNECTO─┬─┘
               ├─NOWNERTO─┤
               └─NONE─────┘
```
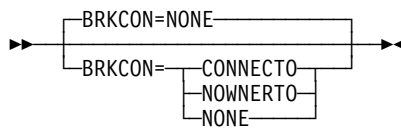
Use: Switched subarea support
Default: NONE

Specifies that NCP is to break a switched connection if a line has been idle (not sending or receiving I-frames) for the period of time specified by ACTIVTO on the GROUP definition statement. BRKCON is valid only if you code DIAL=YES and PUTYPE=4 for switched subarea support lines.

If you code BRKCON=CONNECTO, activity time-out is started at connection time. If you code BRKCON=NOWNERTO, activity time-out is started when the owning SSCP undergoes ANS. If you code MONLINK=CONTINUOUS for a switched subarea line, the SSCP monitor mode function (SMMF) will own the line and will not be subject to ANS. Therefore, BRKCON=NOWNERTO is the same as BRKCON=NONE.

CASDL
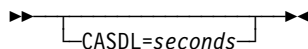
```
►►──────────────────────►◄
     └─CASDL=seconds─┘
```

Use: ESCON logical link stations
Default: Value of TIMEOUT on this PU definition statement.
Minimum: 0.0
Maximum: 840.0

Defines the channel adapter slowdown limit, that is, how long this channel link can block inbound data before signaling that this station is inoperative. You can code CASDL on the PU definition statement only when you are defining a station for an ESCON logical link.

CASDL is not valid if you code TIMEOUT=NONE on this PU definition statement.

COMRATE
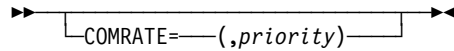
```
►►──┬──────────────────────────┬──►◄
    └─COMRATE=──(,priority)─────┘
```

Use: Frame-relay logical stations (FRTEs)
            Frame-relay physical stations (FHSPs, IPSPs)
            NTRI logical FHSPs
            NTRI physical stations
            ISDN logical stations
Default: COMRATE=(,1)

COMRATE specifies the relative transmit priority for a TESP, FHSP, IPSP, or NTRI physical station.

priority

Use: Frame-relay logical stations (FRTEs)
            Frame-relay physical stations (FHSPs, IPSPs)
            NTRI logical FHSPs
            NTRI physical stations
            ISDN logical stations
Minimum: 1
Maximum: 64
Default: 1

Specifies the relative transmit priority of a TESP, FHSP, or IPSP. The value of the *priority* suboperand is multiplied by the value of the DATABLK keyword on the physical LINE definition statement (*priority*∗DATABLK) to determine the guaranteed minimum number of bytes that a resource can transmit before a different resource must be given a transmit opportunity.

For NTRI physical stations, each defined physical station represents a specific protocol (for example, SNA or IP) as specified by the NETWORK keyword (see NETWORK on the PU definition statement on page 2-401 for more information). The value of the priority suboperand is multiplied by the value of the DATABLK keyword to establish the guaranteed minimum number of bytes provided for that protocol for the transmission of associated traffic before a different protocol is provided a transmission opportunity.

3746 Model 900 Attached Frame-Relay and ISDN Lines: See the CIR keyword on the LINE statement for an alternative way of managing the bandwidth.

DATMODE

```
►►──┬────────────────────────┬──►◄
    └─DATMODE=──┬─FULL─┬──────┘
                └─HALF─┘
```
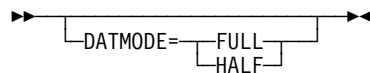
Use: Nonswitched SDLC peripheral link stations
Default: See text

Specifies whether NCP is to use half-duplex or full-duplex scheduling logic when NCP is a primary link station communicating with this adjacent link station. DATMODE is irrelevant if NCP is a secondary link station.

Code DATMODE=HALF if the second suboperand of ADDRESS on the associated LINE definition statement specifies HALF.

It is important to distinguish between DATMODE specified on the PU definition statement, the *coded* DATMODE, and DATMODE used in operation of the link, the *operational* DATMODE. If DATMODE is not specified on the PU definition statement, the *coded* DATMODE defaults to HALF.

Link Stations in Adjacent Type 1 and Type 2.0 Nodes: The *coded* DATMODE is the *operational* DATMODE.

Link Stations in Adjacent Type 2.1 Nodes: In addition to the *coded* DATMODE, DATMODE is supplied in the format-3 XID exchange. The *operational* DATMODE is FULL if both DATMODE supplied in the format-3 XID exchange is FULL and the second suboperand of ADDRESS on the associated LINE definition statement specifies FULL. Otherwise, the *operational* DATMODE is HALF. In either case, the *coded* DATMODE is ignored, because the *DATMODE* is resolved from the format-3 XID exchange. Therefore, the *operational* DATMODE might not be the same as the *coded* DATMODE.

Switched SDLC Peripheral Link Stations: Do not code DATMODE on a PU definition statement when DIAL=YES is coded on the GROUP definition statement. When the peripheral link station in the adjacent node is connected using switched procedures, VTAM passes the *coded* DATMODE value to NCP when a switched connection is established between NCP and the adjacent link station. DATMODE is defined on a VTAM PU definition statement in a VTAM switched major node, whether defined or created dynamically. DATMODE is assumed to be HALF when the second suboperand of ADDRESS on the associated LINE definition statement specifies HALF. All the considerations for DATMODE as defined on a PU definition statement for a nonswitched line apply to the DATMODE passed by VTAM.

DELAY

```
          ┌─DELAY=0.1───────┐
►►────────┼─────────────────┼────►◄
          └─DELAY=seconds───┘
```
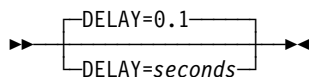
Use: ESCON logical link stations
Default: 0.1
Minimum: 0.0
Maximum: See text

Specifies, in seconds, the maximum interval that NCP is to delay from the time data is available to the host to the time the attention signal is presented to the channel. You can code DELAY on the PU definition statement only when you are defining a station for an ESCON logical link.

The maximum is one-half the value of TIMEOUT on this PU definition statement. If you code TIMEOUT=NONE, the maximum is 420.

DLCI

```
►►──┬──────────────────────────────────────────┬──►◄
    └─DLCI⁽¹⁾=(──address──,──┬──D──┬──)─┘
                             ├──H──┤
                             └──DD─┘
```
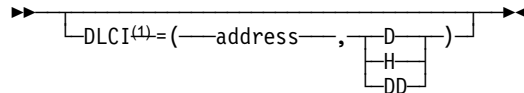
[1] Valid with VTAM V4R3 or later releases

Use: Frame-relay stations; NTRI frame handler logical stations
Default: See text
Minimum: (10,H)
Maximum: See text

DLCI specifies the DLCI of the subport or internet host. All DLCI values associated with a single frame-relay physical line or a NTRI frame handler logical line must be unique. For example, if you use DLCI X'30' on a frame-relay physical line to identify a subarea TESP or a FHSP, address X'30' cannot be used by a peripheral device to establish an incoming logical peripheral connection to NCP over that same physical line.

You should avoid using DLCI X'20' (decimal 32) when you define frame-relay peripheral links, since peripheral devices often use DLCI X'20' for incoming calls. When a subarea link TESP or FHSP uses this DLCI, the peripheral devices must be reconfigured to use a different DLCI.

Although DLCI values are coded in hexadecimal in an NCP generation definition, other number bases may be used for DLCI values defined by other products. For example, default DLCI values defined by the IDNX are decimal numbers.

DLCI definition format and definition rules are as follows:

- DLCI=(dlci,D) (D means DLCI is specified in decimal)
- DLCI=(dlci,H) (H means DLCI is specified in hexadecimal)
- DLCI=(dlci,DD) (DD means DLCI is specified in dotted decimal)

If no type is specified, decimal is assumed.

If the LMI protocol is not supported for this line (LMI=NO on the first PU definition statement for this line), the maximum value for DLCI, if not limited by the MAXDLCI keyword, is the largest DLCI by the line adapter for this line. The maximum value is 991.

If an LMI protocol is supported for this line (LMI is coded and is not equal to NO on the first PU definition statement for this line), the maximum value for DLCI is the lower of the following two values:

- The largest DLCI allowed by the line adapter for this line (see preceding text)

- The value of the following expression:

    $$((maxframe - n) / 5) + 15$$

  where:

    $maxframe$ is the value of MAXFRAME for this line.

    n is 17 when LMI=CCITT or LMI=CCITT-NUI on the LMI PU definition statement.

n is 18 when LMI=ANSI or LMI=ANSI-NUI on the LMI PU definition statement.

For more information, refer to the discussion of how to define frame-relay subarea links in the *NCP, SSP, and EP Resource Definition Guide*.

Frame-Relay Physical Lines: For frame-relay physical lines, DLCI is not valid on the first PU definition statement for this line (the link-station subport). On all subsequent PU definition statements (frame-relay frame handler subports), you can specify the DLCI address using either ADDR or DLCI.

Frame-Relay Logical Peripheral Lines: DLCI is not valid for frame-relay logical peripheral lines.

Frame-Relay IP Resources: You can specify either ADDR or DLCI for frame-relay IP resources.

NTRI Frame Handler Logical Lines: For NTRI frame handler logical lines, DLCI is not valid on the first PU definition statement for this line (the link-station subport). On all subsequent PU definition statements (frame-relay frame handler subports), you can specify the DLCI address using either ADDR or DLCI.

DYNWIND
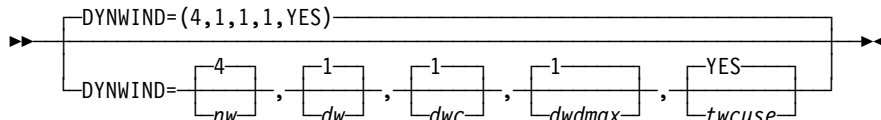
```
        ┌─DYNWIND=(4,1,1,1,YES)──────────────────────────────────┐
►►──────┤                                                        ├──►◄
        └─DYNWIND=──┬─4────┬─,─┬─1──┬─,─┬─1───┬─,─┬─1──────┬─,─┬─YES────┬─┘
                    └─nw───┘   └─dw─┘   └─dwc─┘   └─dwdmax─┘   └─twcuse─┘
```

Use: Token-Ring, frame-relay, and ISDN physical stations for associated
            logical peripheral stations
            ISDN physical stations for ISDN logical subarea stations
            Token-Ring and frame-relay logical subarea stations
Default: See text

Specifies the values to be used by NCP to modify the IEEE 802.2 LLC working window parameter. The values specified increment the size of the working window, or control how quickly the working window should be closed or dropped when frame loss occurs. When you code DYNWIND on a physical line definition, the values specified apply to all logical lines associated with that physical line.

DYNWIND is not valid for FHSP and IP stations.

nw

Minimum: 1
Maximum: 8
Default: 4

Specifies the number of frames that must be received and acknowledged by the adjacent device before the sending NCP increments the working window value by 1. When the sending NCP receives acknowledgments for *nw* consecutive I-frames, the working window value is incremented, until it reaches the value specified by MAXOUT for leased connections, or the value specified in the VTAM switched major node for switched connections.

dw

Default: 1

Specifies the value by which the sending NCP divides or decreases the working window value when frame loss occurs.

1 is a special value which indicates that the working window value should be set to 1 when frame loss is detected. Valid **divisor values** are 2, 4, 8, or 16. Starting in NCP V7R6, valid **percentage values** are 6, 12, 25, 50, 75, 87, or 93. Table 2-102 summarizes the valid *dw* values.

dw is valid only for frame-relay and ISDN lines.

dwc

Default: 1

Specifies the value by which the sending NCP divides or decreases the working window value when backward explicit congestion notification (BECN) occurs.

1 is a special value which indicates that the working window value should be set to 1 when BECN is detected. Valid **divisor values** are 2, 4, 8, or 16. Starting in NCP V7R6, valid **percentage values** are 6, 12, 25, 50, 75, 87, or 93. Table 2-102 summarizes the valid *dwc* values.

dwc is valid only for frame-relay and ISDN lines. See the *NCP and EP Reference* for more information about BECN.

Table 2-102. Working Window Reductions for dw and dwc Values

	Value	Percent Reduction
Special Value	1	N/A
Divisor Values	2	50
	4	75
	8	87
	16	93
Percentage Values	6	6
	12	12
	25	25
	50	50
	75	75
	87	87
	93	93

Note: The default values for *dw* and *dwc* cause the dynamic window algorithm to work according to IEEE 802.2 architecture specifications. Using other valid values allows you to increase bandwidth utilization.

dwdmax

Use: 3745 frame-relay lines
Default: 1 second
Minimum: 0.1 seconds
Maximum: 6 seconds

Specifies the upper limit for the time delay that NCP initiates before transmitting the next I-frame for an FRTE station experiencing congestion. NCP initiates this time delay if congestion continues while a station's working transmit window is equal to 1. If you do not want NCP to initiate the delay, specify a value of 0.

twcuse

Use: 3745 frame-relay lines
Default: YES

Specifies whether or not an NCP FRTE station ignores any other BECN for the window congestion time of 100 milliseconds (ms) after receiving the initial BECN. Specify YES to have a station ignore subsequent BECNs for 100 ms if the working transmit window is adjusted or NO to have the stations react to all BECNs. This suboperand is intended to keep NCP stations from overreacting to mild congestion.

| **3746 Model 900 Attached Frame-Relay and ISDN Lines:**   See the CIR keyword
| on the LINE statement for an alternative way of managing the bandwidth.

ERRORT
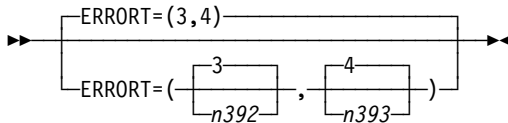
```
         ┌─ERRORT=(3,4)──────────────────┐
►►───────┤                               ├──────►◄
         │           ┌─3─┐     ┌─4─┐     │
         └─ERRORT=(──┤   ├──,──┤   ├──)──┘
                     └n392┘     └n393┘
```

Use: Frame relay (link-station subport only)
            NTRI frame handler logical line (link-station subport only)
            ISDN physical lines and stations
Default: (3,4)
Minimum: 1
Maximum: 10

Specifies the LMI error threshold at which NCP determines that communication with an adjacent frame-relay device has been lost. The LMI error threshold is reached when *n392*[12] LMI errors occur in *n393*[12] consecutive LMI events. The value you specify for *n392* must be less than or equal to the value you specify for *n393*.

ERRORT is valid only when this subport supports LMI protocol, as specified by the LMI keyword (not LMI=NO).

Note: For NCP V7R2 and earlier releases, NCP will issue a +RSP(IPLFINAL) to VTAM to indicate that the load is complete. Because the remote NCP cannot respond to LMI messages during the load process, frame-relay physical lines attached to the remote NCP may become inoperative (INOP 02) after the load is complete. In this event, you need to reactivate the physical lines. For NCP V7R3 and later releases, the physical line no longer INOPs. The logical FRTE line may become inoperative (INOP 02)

---

[12] n392 and n393 are standard frame-relay values described by CCITT standard Q.933 Annex A and ANSI standard T1.617 Annex D.

and the LMI link-station may also become inoperative (INOP 01). In this
event, VTAM should reactivate the resources.

To avoid the condition where an INOP occurs during a remote load, the LMI
error threshold time interval for the link station subport should be at least 3
times the T1 timer interval for the logical subarea line. The LMI error
threshold time interval is $t391 \times n392$, where $t391$ is a time interval defined
by TIMERS on the PU definition statement for the link station subport.

If you do not code the time intervals as just described, the LMI PU and
associated logical lines over which NCP is being loaded may become inop-
erative (INOP 01 and INOP 02) respectively *before* NCP issues the
+RSP(IPLFINAL) to VTAM, that is, before the load is complete. In this
event, the load will fail.

See the *NCP, SSP, and EP Resource Definition Guide* for more information
about how to code the ERRORT keyword when the line will be used for
remote loading of an NCP.

FRSPRI

```
        ┌─FRSPRI=0─┐
►►──────┤          ├──────►◄
        └─FRSPRI=n─┘
```

Use: 3746 Model 900 congestion control
Default: 0
Minimum: 0
Maximum: 3

FRSPRI specifies a relative service priority at a station level. The lowest priority is
0, and the highest priority is 3.

When congestion occurs on a 3746 Model 900 attached Frame Relay/ISDN phys-
ical link, the 3746 Model 900 will use the relative service priority to determine the
order in which stations will be serviced during the next transmit period.

The following section lists the stations that the FRSPRI parameter supports:

- Frame Relay Frame Handler PUs
- Frame Relay Logical Subarea PUs
- Frame Relay Logical Peripheral PUs
- ISDN Logical Subarea PUs (LOGical,SUB)
- ISDN Logical Peripheral PUs (LOGical,PER)

See the *3745 Communication Controller Models A, 3746 Expansion Unit Model
900: Migration and Planning Guide*, GA33-0183, for more information about the
Model 900 congestion control processing.

GP3174

```
        ┌─GP3174=NO──────────────┐
►►──────┤                        ├──────►◄
        └─GP3174=─┬─group_address─┬─┘
                  └─NO────────────┘
```
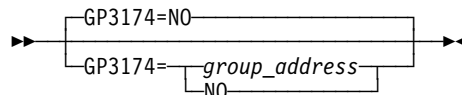
Default: NO

Specifies that a poll to an IBM 3174 gateway controller can be a group poll (group polling is a process that sends a single poll to one collection point for a group of stations, inviting a response from any station in the group that has data to send) for any station on the IBM 3174, including the IBM 3174 and the stations attached to the Token Ring.

group_address indicates any two hexadecimal digits except X'00' and X'FF'. If you code LPDATS=LPDA2, the hexadecimal digit X'FD' is not valid. *group_address* is valid if you specify LNCTL=SDLC and DIAL=NO on the GROUP definition statement, and PUTYPE=2 and MODULO=8 on the PU definition statement. You may specify a maximum of 16 different group addresses for each line. *group_address* must be different from any address specified on ADDR for each physical unit on a line.

See your controller documentation for any product specific addressing considerations.

HPRQLIM
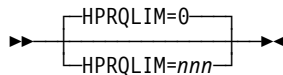
```
        ┌─HPRQLIM=0───┐
►►──────┤             ├──────►◄
        └─HPRQLIM=nnn─┘
```

Use: Composite ANR node's subarea network flow control
Default: 0
Minimum: 0
Maximum: 16777215

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 its 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.

This keyword is supported only when PUTYPE=2 and XID=YES.

INNPORT

```
►►──────────┬──────────────────────┬──────────►◄
            └─INNPORT=─┬─NO──┬──────┘
                       └─YES─┘
```
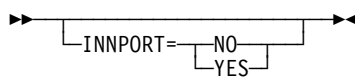
Use: Frame relay, ISDN, Token Ring, Ethernet-type LANs
Default: See text

Specifies whether a physical unit will remain active when CDLINK=ACT is specified on either the HALT or VARY INACT commands. Code INNPORT=YES if you do not want the physical unit deactivated.

The INNPORT keyword is used for type 1 physical units that represent physical connections carrying SNA subarea or non-SNA routed traffic, such as internet or frame-relay frame handler data.

You can code INNPORT on the PU definition statement when you:

- Code PHYPORT or PHYSRSC on a subarea logical GROUP definition statement
- Define Ethernet-type LAN physical units
- Define Token-Ring physical units with NETWORK=IP

NDF adds INNPORT=YES to all PU definition statements that follow GROUP definition statements where:

- ECLTYPE=PHYSICAL and ADAPTER=TIC2 or TIC3
- ECLTYPE=(PHYSICAL,ANY)
- ECLTYPE=(PHYSICAL,SUBAREA)
- ETHERNET=PHYSICAL
- FRELAY=PHYSICAL
- FRELAY=(PHYSICAL,ANY)
- FRELAY=(PHYSICAL,SUBAREA)
- ISDN=PHYSICAL
- ISDN=(LOGICAL,B)

INTFACE

```
►►──┬─────────────────────────────────┬──►◄
    └─INTFACE=(──┬──────┬──,──┬─────┬──)─┘
                 └─name─┘      └─mtu─┘
```
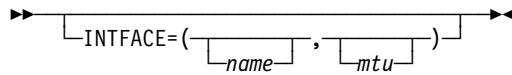
Use: Frame-relay IP interfaces, channel IP interfaces, and internal NCP to Model 900 point-to-point IP connections

Specifies an internet router interface name, which must be unique within your NCP generation. If a corresponding IPLOCAL definition statement is defined, the interface name defined must be the same as specified on the PU definition statement.

Internal NCP to Model 900 Point-to-Point IP Connections: When the PU statement omits either the INTFACE *name* suboperand or the entire INTFACE keyword, the gen will expect to find the IP interface parameters coded on the PU statement, rather than on an associated IPLOCAL statement.

name

Default: None

Specifies the interface name, which must be unique and no more than 8 characters long.

When IPLOCAL keywords (for example, LADDR) are specified on the PU definition statement and no corresponding IPLOCAL definition statement is specified, the PU label name is used as the interface name instead of the name parameter.

mtu

Default: See text
Minimum: See text
Maximum: See text

Specifies the maximum transfer unit, which is the maximum size, in bytes, of the transmission frame. This suboperand is not valid for IP channel connections.

Frame-Relay IP Interfaces: The default is either 2048 or the value of MAXFRAME-14 (X'E') on the LINE statement, whichever is smaller. The minimum is 282. Ensure that the value you specify does not exceed the value of MAXFRAME on the LINE statement.

Internal NCP to Model 900 Point-to-Point IP Connections: When you code NETWORK=IP on this PU definition statement and ECLTYPE=PHYSICAL on the LINE definition statement preceding this PU definition statement, the default value for *mtu* is 2048 or the value coded for MAXTSL minus 4, whichever is less. The minimum value is 68, and the maximum value is MAXTSL minus 4.

IPQLIM

```
►►──┬────────────────┬──►◄
     └─IPQLIM=nnn─────┘
```
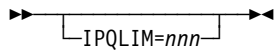
Use: Frame-relay internet resources
Default: See text
Maximum: 65535

Specifies the maximum amount of data that can be queued on a frame-relay internet outbound queue awaiting transmission. The minimum value is the
| MAXFRAME value specified on the LINE definition statement. For NCP V7R6 and
| earlier releases, the default value is the MAXFRAME value specified on the LINE
| definition statement. Starting in NCP V7R7, the default value is 3 times the
| MAXFRAME value.

IRETRY

```
              ┌─IRETRY=NO────────┐
►►──┼──────────────────────┼──►◄
              └─IRETRY=─┬─YES─┬──┘
                        └─NO──┘
```
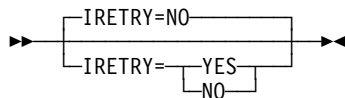
Use: Nonswitched links
Default: NO

Specifies whether NCP will repoll a given secondary station before advancing to the next station in the SOT. If the primary station detects a time-out during a poll cycle, IRETRY=YES causes NCP to repoll the secondary station before advancing to the next station in the SOT. While errors persist, NCP continues to poll a given station up to *m* times, where the number of repolls is determined by the value of *m* specified for the RETRIES keyword on the LINE definition statement.

IRETRY is valid only if you specify DIAL=NO on the GROUP definition statement.

For casual connection, code IRETRY=NO when MODE=SECONDARY on the GROUP definition statement.

Ethernet-Type LANs: If you code IRETRY for an Ethernet-type LAN connection, you must code IRETRY=NO.

ISDNCFG

```
                  ┌─ISDNCFG=CRC─────────────────┐
►►────────────────┤                             ├──►◄
                  └─ISDNCFG=──┬─CRC───┬──────────┘
                              └─NOCRC─┘
```
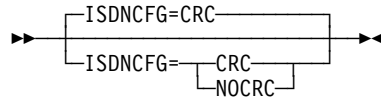
Use: ISDN physical lines
Default: CRC

Specifies the ISDN channel configuration, specifically whether cyclic redundancy check (CRC) is included in frames to detect frame loss.

ISDNDSCT

```
                  ┌─ISDNDSCT=15.0───────────────┐
►►────────────────┤                             ├──►◄
                  └─ISDNDSCT=──┬─seconds─┬───────┘
                               └─NO──────┘
```
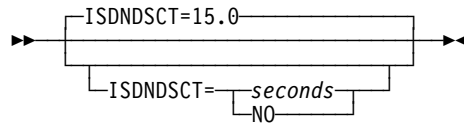
Use: ISDN physical lines
Default: 15.0
Minimum: 1.0
Maximum: 65535.0

Specifies how long a 3746 Model 900 is to wait to bring down the ISDN B-channel after the last associated logical station disconnects (ISDNDCST=*seconds*) or that the ISDN B-channel should remain active even when the ISDN B-channel does not have any associated logical stations (ISDNDSCT=NO).

Specify *seconds* either as an integral number of seconds or to the nearest 10th of a second.

LADDR
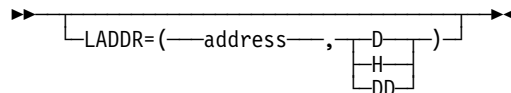
```
►►──┬────────────────────────────────────────┬──►◄
    └─LADDR=(──address──,──┬─D──┬──)──────────┘
                           ├─H──┤
                           └─DD─┘
```

Use: Frame-relay IP resources, channel IP resources, and internal NCP to
            Model 900 point-to-point IP connections
Default: None

Specifies the internet address of the associated interface.

address is a 4-byte internet address that may be specified in either dotted decimal or hexadecimal format. The value you specify for LADDR cannot be specified on any other PU or IPLOCAL definition statement. The address specified must be in the same network or subnetwork as the locally attached network or subnetwork. The network and subnetwork portions of the LADDR address must match the address of the associated network or subnetwork.

To code the address as a decimal number, use the format LADDR=(*dddddddddd*,D) (D means decimal representation). For example, code the address 128.255.16.64 as LADDR=(2164199488,D).

Note: The value 2147483648 (2^{31}) is not valid.

To code the address as a hexadecimal number, use the format LADDR=*hhhhhhhh*
or LADDR=(*hhhhhhhh*,H) (H means hexadecimal representation). For example,
code the address 128.255.16.64 as LADDR=80FF1040 or LADDR=(80FF1040,H).

To code the address as a dotted decimal number, use the format
LADDR=*n0.n1.n2.n3* or LADDR=(*n0.n1.n2.n3*,DD) (DD means dotted decimal rep-
resentation) where *n0–n3* are numbers between 0 and 255. For example,
LADDR=128.255.16.64 or LADDR=(128.255.16.64,DD).

The address specified by LADDR is associated with this NCP and can be a desti-
nation internet address for certain Transmission Control Protocol/Internet Protocol
(TCP/IP) functions. For more information, see "IP Datagram Fragmentation and
Reassembly" in Chapter 7 of the *NCP and EP Reference*.

See the *NCP, SSP, and EP Resource Definition Guide* for more information about
internet addresses.

LANTYPE

```
►►──────────────────────────────────►◄
     └─LANTYPE=─┬─ENETV2──┬─
               ├─802.3───┤
               └─DYNAMIC─┘
```

Use: Ethernet-type LANs
Default: None

Required for Ethernet-type LANs unless you code LANTYPE on the GROUP or
LINE definition statement. Specifies the type of frame format that this physical unit
supports.

ENETV2
> Specifies that this physical unit supports the Ethernet Version 2 frame format
> only.

802.3
> Specifies that this physical unit supports the IEEE 802.3 standard frame format
> only.

DYNAMIC
> Specifies that this physical unit supports both the Ethernet Version 2 and IEEE
> 802.3 frame formats concurrently. You should not specify
> LANTYPE=DYNAMIC unless this physical unit will actually carry both frame
> formats. NCP performance is slightly lower for LANTYPE=DYNAMIC because
> NCP performs certain processing for both formats.

If you code LANTYPE=DYNAMIC, you must also code LANTYPE on the IPHOST
definition statement. If you code LANTYPE=ENETV2 or 802.3, NCP uses this
value as the default for all internet hosts attached to this Ethernet-type LAN.

LLERP
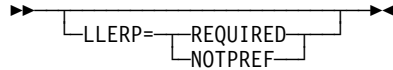
```
►►──┬──────────────────────────┬──►◄
     └─LLERP=─┬─REQUIRED─┬──────┘
              └─NOTPREF──┘
```

Use: Token-Ring, frame-relay, and ISDN physical stations for associated
            logical peripheral stations
            ISDN physical stations for associated logical subarea stations
            Token-Ring and frame-relay logical subarea stations
Default: See text

LLERP specifies whether the link station will perform link level error recovery proce-
dures (ERP) when sending HPR data on this link station. Specify
LLERP=NOTPREF if you want NCP to bypass the error recovery procedures; it is
best to bypass link level ERP when transmitting HPR data over highly reliable lines.
Specify LLERP=REQUIRED only if the line quality is poor, and the adjacent node
supports the use of link level ERP for HPR data.

NCP supports the use of link level ERP for HPR data on all link stations which
support HPR. NCP supports sending HPR data without using link level error
recovery only on the following link station types:

- Frame-relay logical subarea
- Frame-relay logical peripheral
- NTRI logical peripheral
- 3746 Model 900 Token-Ring logical subarea
- 3746 Model 900 Token-Ring logical peripheral
- ISDN logical subarea
- ISDN logical peripheral

Link level ERP support for HPR data is defined on the physical port definition: on
the frame-relay LMI PU, on the ISDN physical PU, or on the Token-Ring physical
PU for SNA (the one with NETWORK=SNA). The logical connections that map to
the physical port will inherit that physical's defined (or defaulted) support for
LLERP. If LLERP=NOTPREF was coded on the physical port, you may override
that by coding LLERP=REQUIRED on the logical definitions as follows:

Frame-Relay Logical Subarea: Code LLERP=REQUIRED on the logical subarea
     PU statement in the NCP definition deck.

3746 Model 900 Token-Ring Logical Subarea: Code LLERP=REQUIRED on the
     logical subarea PU statement in the NCP definition deck (starting with NCP
     V7R4).

Frame-Relay LMI PU: The default is NOTPREF.

Frame-Relay Logical Subarea PU: The default is NOTPREF.

NTRI Token-Ring Physical PU for SNA: The default is NOTPREF.

3746 Model 900 Token-Ring Physical PU: The default is NOTPREF.

3746 Model 900 Token-Ring Subarea PU: The default is NOTPREF.

ISDN Physical PU: The default is NOTPREF.

LMI
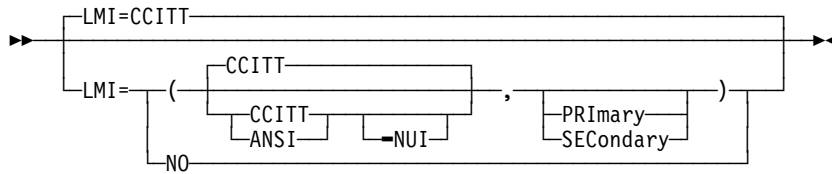
```
                            ┌─LMI=CCITT───────────────────────────────────┐
   ►►─────────────────────────────────────────────────────────────────────────►◄
          │           ┌─CCITT─┐                             │
          └─LMI=─┬─(─┬─────────┬──────────┬────────,──┬──────────────┬──)─┘
                 │   │ ┌─CCITT─┐│   ┌─-NUI─┐        ├─PRImary────┤
                 │   └─┴─ANSI──┴┘   └──────┘        └─SECondary──┘
                 └─NO─────────────────────────────────────────────
```

Use: Frame relay (link-station subport only)
           ISDN (physical PU only)
           NTRI frame handler logical link (link-station subport only)
Default: CCITT

Defines the LMI protocol standard and NCP echo detection for this subport.

The LMI keyword is valid only for the link-station subport, that is, the first PU defi-
nition statement under a LINE definition statement in a group of frame-relay or
ISDN physical lines or NTRI frame handler logical lines.

Note: The maximum valid value for ADDR may be reduced when you include LMI
protocol support. See DLCI on the PU definition statement on page 2-377
for more information.

CCITT│ANSI
Specifies that this frame-relay or ISDN subport is to support LMI protocol and
specifies which LMI standard NCP is to use for this subport. If you specify
CCITT, NCP uses CCITT Q.933 Annex A, the international standard. If you
specify ANSI, NCP uses ANSI T1.617 Annex D, the U.S. standard. Use the
same LMI standard for adjacent nodes.

LMI protocol provides information on permanent virtual circuit (PVC) status and
link integrity. Use LMI protocol for this subport if the adjacent subport supports
LMI protocol. To facilitate migration, you can code LMI=CCITT or LMI=ANSI if
the adjacent subport is running NCP V6R1 even though NCP V6R1 does not
support LMI protocol. When you do this, this subport will operate as though
you coded LMI=NO. Code LMI=NO in all other cases.

-NUI

The -NUI (Network-to-User Interface) on the LMI-standard parameter indicates
that only the Network-to-User LMI protocol will be used on this frame relay
interface. When -NUI is specified (for example, LMI=CCITT-NUI), no discovery
of the LMI protocol will be done.

Attention: If the adjacent frame relay device is also only NUI-capable, speci-
fying -NUI will result in the frame-relay interface remaining in an activation
pending state.

Use of -NUI speeds activation of the frame-relay interface when the adjacent
frame-relay device is only UNI- (User-to-Network interface) capable by elimi-
nating the time (default of 40 seconds) it takes to determine that the adjacent
device is UNI-only. This might be a consideration for switched frame-relay
interfaces (switched frame-relay physical lines or ISDN) that might be used for
only a short duration.

PRImary│SECondary
> Specifies that this subport is to support the NCP echo detection function and specifies which echo detection operating mode NCP is to use. (**Note:** Echo detection is an NCP function; it is not a frame-relay standard.)

> When NCP transmits a frame to a satellite, NCP may interpret an echo from the satellite as an incoming frame. To prevent this, NCP echo detection sets the Command/Response (C/R) bit in all outbound frames to 1 (PRIMARY) or 0 (SECONDARY) and rejects any inbound frames whose C/R bit is the same as the C/R bit in outbound frames.

> **Note:** Echo detection is not supported for NTRI frame handler logical lines and ISDN physical lines. As a result, neither PRIMARY nor SECONDARY is valid and either is ignored if specified.

> To use echo detection, specify PRIMARY or SECONDARY as follows:

> - If the subports on both ends of this line support NCP echo detection, specify PRIMARY for one subport and SECONDARY for the other. It does not matter which subport you define as PRIMARY and which you define as SECONDARY.

> - If the C/R bit is unused (that is, always set to 0) by the subport on the other end of this line, specify PRIMARY for this subport.

NO
> Specifies that this port is not to support LMI protocol.

LMODADR

```
►►─┬─────────────────────────────────┬─►◄
   └─LMODADR=(address1─┬──────────┬─)─┘
                       └─,address2─┘
```
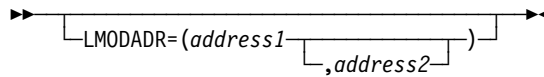
Default: See text

Specifies the local modem addresses used on the line going to the physical unit. LMODADR is valid on the PU statement only if you specify LPDATS=LPDA2 on the LINE definition statement.

address1 and *address2* must be hexadecimal values between X'0' and X'F'. *address1* specifies the local modem address on link segment 1; *address2* specifies the local modem address on link segment 2. *address2* is valid only if SEGMENT=2. The default value is LMODADR=1 if SEGMENT=1; the default value is LMODADR=(1,2) if SEGMENT=2.

LOCALTO

```
►►─┬──────────────────┬─►◄
   └─LOCALTO=seconds─┘
```
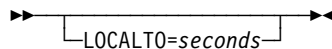
Use: Token-Ring logical subarea and frame-relay logical resources
Default: See text
Minimum: 0.6
Maximum: 20.0

Specifies the reply timer (T1) value in seconds. You can also specify this value using T1TIMER coded on the following definition statements:

- BUILD for Token-Ring logical and peripheral resources and NTRI frame handler logical resources
- LINE for Token-Ring logical subarea resources
- LINE for frame-relay logical resources

Token Ring: LOCALTO specifies the Token-Ring reply timer (T1) value for stations connected to a local Token Ring. LOCALTO is valid only for logical subarea lines.

The value of LOCALTO must not be greater than the value of REMOTTO on this LINE definition statement. The default value for LOCALTO is either the value of REMOTTO on this LINE definition statement or the value of LOCALTO on the BUILD definition statement, whichever is less.

Frame Relay: LOCALTO specifies the frame-relay reply timer value for stations connected to a frame-relay network. If you do not code LOCALTO on either the LINE or GROUP definition statement, the default is 6.0. The value of LOCALTO on the BUILD definition statement is not used for frame relay.

Important: Code a value for LOCALTO that reflects the maximum amount of time that it takes to receive a solicited response from the adjacent link station. Factors that can affect this time include line speed, propagation delay, maximum frame size, delay caused by intermediate frame handlers, and the number of DLCIs supported by the associated physical line that could create queuing delays for the FRTEs. If the value you code for LOCALTO is too small, the FRTEs can lose synchronization and fail the connection. This is indicated by an INOP 01 and a link event or an alert that indicates a frame reject (FRMR) condition. If this occurs, code a larger LOCALTO value for this FRTE connection.

If a substitute route for a PVC is defined in the frame-relay network, you may need to code the values of LOCALTO and RETRIES such that the combination of these two values is greater than the LMI error threshold of any node in the PVC. This gives the network time to switch to the substitute PVC.

An indication of INOP 02 can occur for a frame-relay logical line if it is used to perform a remote load. To avoid the condition where the INOP 02 occurs prior to the +RSP(IPFINAL), the LMI error threshold time interval for the link station subport should be at least 3 times the T1 timer interval for the logical subarea line. The LMI error threshold time interval is $t391 \times n392$, where $t391$ is a time interval defined by TIMERS on the PU definition statement for the link station subport, and $n392$ is an error count threshold defined by ERRORT on the same PU definition statement.

LPDA

```
►►──┬─────────────────────┬──►◄
    └─LPDA=─┬─BLOCK─┬──────┘
            └─ALLOW─┘
```

Default: See text

Specifies whether NCP is to run LPDA tests for this physical unit. Code BLOCK if you do not want to run LPDA tests; code ALLOW if you want to run LPDA tests.

You can block or allow LPDA testing separately on the primary and secondary sides of a subarea link.

If you code LPDATS=LPDA1 or LPDATS=LPDA2 on the LINE definition statement, the default is LPDA=ALLOW. For more information on LPDA testing, see LPDATS on the LINE definition statement on page 2-240.

For casual connection, LPDA is not valid if MODE=SECONDARY is coded on the GROUP definition statement.

Ethernet-Type LANs: If ETHERNET=PHYSICAL is specified on the GROUP defi-nition statement, only LPDA=BLOCK is valid. If you do not code LPDA, NDF adds LPDA=BLOCK to the generation definition.

MAXBFRU

```
              ┌─MAXBFRU=32───┐
►►─┬──────────────────┬─►◄
   └─MAXBFRU=count─┘
```
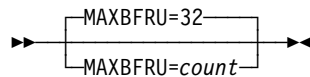
Default: 32
Minimum: 1
Maximum: 255

Specifies the number of buffer units the access method allocates for receiving data from NCP. MAXBFRU is valid only when LNCTL=CA and CA=TYPE7 or CA=TYPE7-TPS. See TRANSFR on the LINE definition statement on page 2-286 for more information on the maximum amount of data VTAM can receive in one data transfer.

ESCON Links: For ESCON logical link stations, it is recommended that you code MAXBFRU with a value equal to $10 \times len \div$ UNITSZ, where len is the average PIU length and UNITSZ is the size of the VTAM channel read buffer (IOBUF), as defined by UNITSZ on the VTAM HOST definition state-ment. The value of MAXBFRU times UNITSZ must not exceed 64KB.

MAXBFRU is not valid for ESCON physical link stations.

MAXDATA
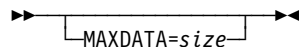
```
►►─┬──────────────┬─►◄
   └─MAXDATA=size─┘
```

Use: Nonswitched link stations
Default: See text
Minimum: See text
Maximum: 65535

Specifies the size of the largest path information unit (PIU) or PIU segment that the adjacent link station can receive. An unsegmented PIU consists of a transmission header (TH), a request header (RH), and request unit (RU) data. A segmented PIU consists of a TH and segmented RU data. The first segment of a sequence also contains the RH.

Note: MAXDATA on the PU definition statement and MAXDATA on the VTAM-only PCCU definition statement have different functions.

It is important to distinguish between:

- The value of MAXDATA as specified on the PU definition statement, which is the *coded* value of MAXDATA.

- The value of MAXDATA used in the operation of the link, which is the *operational* value of MAXDATA.

If MAXDATA is not specified on the PU definition statement, any default value is assigned to the *coded* MAXDATA.

Link Stations in Adjacent Type 1 and Type 2.0 Nodes: The *coded* MAXDATA value is the *operational* MAXDATA value. The default value of MAXDATA is the value of BFRS on the BUILD definition statement plus 9. The minimum value of MAXDATA is the value of BFRS on the BUILD definition statement plus 9.

To determine an appropriate value for MAXDATA, see the documentation for the device or programming associated with the adjacent link station you are defining.

Link Stations in Adjacent Type 2.1 Nodes: A value for MAXDATA is supplied in the format-3 XID exchange. The *operational* MAXDATA value is the value supplied in the format-3 XID exchange. The *coded* MAXDATA value is ignored, because the *operational* MAXDATA is resolved from the format-3 XID exchange. Therefore, the *operational* MAXDATA value might be different from the *coded* MAXDATA value.

The minimum value of MAXDATA is the value of BFRS on the BUILD definition statement plus 9. If the value of MAXDATA supplied in the format-3 XID exchange is less than the minimum value of MAXDATA, an XID negotiation error results and the link is not established.

Link Stations in Adjacent Type 4 Nodes: If the link being defined is a member of a mixed media multiple link transmission group (MMMLTG), you might need to ensure that PIUs are segmented in order to use a segment size which will be acceptable for all members of the MMMLTG. Consult the *NCP, SSP, and EP Resource Definition Guide* for more information about coordinating PIU segment sizes with MMMLTGs.

MAXDATA does not require a default because the *operational* MAXDATA value will be determined from other parameters dynamically. The minimum value of MAXDATA is the value of BFRS on the BUILD definition statement plus 26.

Switched SDLC Subarea and ISDN Subarea and Peripheral Link Stations: Do not code MAXDATA on a PU definition statement when DIAL=YES is coded on the GROUP definition statement. When the link station in the adjacent node is connected using switched procedures, VTAM passes the *coded* MAXDATA value to NCP when a switched connection is established between NCP and the adjacent link station. MAXDATA is defined on a PU definition statement, similar to the NCP PU definition statement, in a VTAM switched major node, whether defined or created dynamically. All the considerations for MAXDATA as defined on a PU definition statement for a nonswitched line apply to the value of MAXDATA passed by VTAM.

Frame Relay Logical Subarea: MAXDATA must not exceed the value of MAXFRAME on the LINE definition statement defining the associated frame-relay physical resource.

MAXDLCI
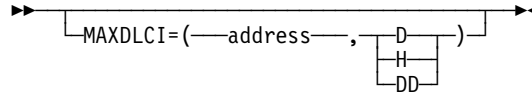
```
►►──┬─────────────────────────────────┬──►◄
    └─MAXDLCI=(──address──,──┬─D──┬──)─┘
                            ├─H──┤
                            └─DD─┘
```

Use: Frame relay (link-station subport only)
            ISDN (physical PU only)
            NTRI frame handler logical link (link-station subport only)
Default: See text
Minimum: X'10'
| Maximum: See Table 2-103 on page 2-394

Frame Relay: MAXDLCI specifies the largest DLCI that can be used for the asso-
| ciated frame-relay interface. The default is the maximum allowed
| value; see Table 2-103 on page 2-394.

ISDN: MAXDLCI specifies the largest DLCI used by all associated
            B-channel lines thereby limiting the DLCI range supported by the
            B-channels.

            Note: There is no default value for ISDN MAXDLCI.

Specifying a MAXDLCI value reduces the number of DLCIs reported in LMI status messages sent to the adjacent mode. MAXDLCI can be specified in three number bases:

- decimal
- hexadecimal
- dotted decimal

If only a DLCI value is specified (such as MAXDLCI=254), then a decimal value is assumed.

MAXDLCI definition format and definition rules are as follows:

- MAXDLCI=(dlci,D) (D means MAXDLCI is specified in decimal representation)
- MAXDLCI=(dlci,H) (H means MAXDLCI is specified in hexadecimal represen-
  tation)
- MAXDLCI=(dlci,DD) (DD means MAXDLCI is specified in dotted decimal repre-
  sentation)

If LMI=NO is specified, the maximum value is 991.

If an LMI protocol is specified for the LMI keyword, the maximum value for MAXDLCI is determined by the following expression:

$$((\textit{maxframe} - n) / 5) + 15$$

where:

maxframe is the value of MAXFRAME for this line.

n is 17 when LMI=CCITT or LMI=CCITT-NUI on the LMI PU definition state-
ment.

n is 18 when LMI=ANSI or LMI=ANSI-NUI on the LMI PU definition statement.

The range of values for MAXDLCI is shown in the following table:

| |

Table 2-103. Values for MAXDLCI on the LINE Definition Statement		
LMI=NO	**LMI=CCITT** **LMI=CCITT-NUI**	**LMI=ANSI** **LMI=ANSI-NUI**
16 to 991	16 to ((MAXFRAME-17)/5) + 15 with an upper limit of 991	16 to ((MAXFRAME-18)/5) + 15 with an upper limit of 991

The range depends on the LMI support specified and the corresponding line's MAXFRAME value.

MAXLU

►►─┬─────────────┬─►◄
   └─MAXLU=*count*─┘
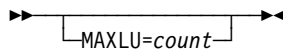
Use: User-written switched lines or user-written dynamic reconfiguration lines
Default: See text
Minimum: 1
Maximum: 255

Specifies the maximum number of logical units that can be associated with the physical unit. MAXLU is valid only if you code VIRTUAL=YES on the GROUP definition statement. If you omit MAXLU, the default for *count* is the number of logical units defined for this physical unit. For a type 1 physical unit, the maximum number of logical units is *count* plus 1.

If dynamic reconfiguration is supported, additional logical units can be added to this physical unit up to *count* minus 1 for a type 1 physical unit and up to *count* for a type 2 physical unit. You cannot dynamically add a logical unit that has a LOCADDR value greater than the value coded for MAXLU for the physical unit. If this physical unit is on a switched link, MAXLU is required.

The value you specify for MAXLU must be equal to or greater than the value specified for PUDRPOOL.

MAXOUT
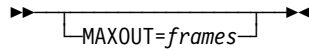
```
▶▶──┬──────────────────┬──▶◀
     └─MAXOUT=frames─┘
```

Use: Nonswitched link stations
Default: See text
Minimum: See text
Maximum: See text

Specifies a limiting value for sending or queueing frames for various types of link station.

For SDLC and Token-Ring and frame-relay TESP logical stations, MAXOUT specifies the maximum number of information frames (I-frames) that can be sent to an adjacent link station without having been acknowledged, in other words, an acknowledgement for the frames which have been sent is expected but has not yet been received.

It is important to distinguish between:

* The value of MAXOUT as specified on the PU definition statement, which is the *coded* value of MAXOUT.

* The value of MAXOUT used in the operation of the station, which is the *operational* value of MAXOUT.

If MAXOUT is not specified on the PU definition statement, any default value is assigned to the *coded* MAXOUT.

SDLC Stations: When a link station in full-duplex mode is linked to an adjacent link station in full-duplex mode, the link stations can send frames simultaneously. Arriving frames from both link stations will be acknowledging frames received from the adjacent link station. However, either link station stops sending frames when the limit defined by the *operational* MAXOUT value is reached and resumes sending frames when acknowledgements for some or all of the frames already sent are received. Nevertheless, a primary link station with *full-duplex* capability can send frames to another secondary link station while waiting for the acknowledgement.

With a link station with *half-duplex* capability, either link station sends frames up to the limit that the MAXOUT *operational* value defines and then requests acknowledgement of the sent frames, regardless of the capability of the adjacent link station. A primary link station is required to send a frame containing the *poll* indicator. A secondary link station is required to send a frame with the *final* indicator. In either case, the link station must then wait for a frame with the acknowledgement to arrive before sending additional frames.

NCP can be either a primary or a secondary link station. The second suboperand of the ADDRESS keyword of the associated LINE definition statement specifies whether NCP is operating *half-duplex* or *full-duplex*, respectively, HALF or FULL.

A link station in an adjacent Type 1 or 2.0 node operates *half-duplex* or *full-duplex* depending on whether the DATMODE keyword of the PU definition statement specifies, respectively, HALF or FULL.

A link station in an adjacent Type 2.1 node operates *half-duplex* or *full-duplex* according to the capability of the link station as described in the format-3 XID exchange.

A link station in an adjacent Type 4 node operates *half-duplex* or *full-duplex* according to the capability of the link station as described in the format-2 XID exchange.

In the case of an adjacent Type 2.1 or 4 node, a secondary adjacent link station operates full-duplex only if the primary adjacent link station also operates *full-duplex*. If the primary adjacent link station operates *half-duplex*, a secondary adjacent link station necessarily operates *half-duplex*.

Whatever other influences there are leading to a determination of the *operational* MAXOUT value, the *coded* MAXOUT value always acts as an upper limit. However, the *direction* to which the *operational* MAXOUT value applies is dependent on the configuration.

A larger *operational* MAXOUT value increases the possibility that arriving frames will acknowledge sent frames where both adjacent link stations are operating *full-duplex* or decreases time-consuming reversals of transmission when at least one of the adjacent link stations is operating *half-duplex*.

A smaller *operational* MAXOUT value decreases a loss of frames when an error on the link between the two adjacent link stations occurs because fewer frames need to be resent and decreases the number of NCP buffers needed to store the frames after sending pending acknowledgement that they have arrived at the adjacent link station.

When the adjacent link station is operating in modulo-8 mode:

- The default value of MAXOUT is:
  - 1 for a link station in a Type 1, 2.0 or 2.1 node
  - 7 for a link station in a Type 4 node
- The minimum value of MAXOUT is 1
- The maximum value of MAXOUT is 7

When the adjacent link station is operating in modulo-128 mode:

- The default value of MAXOUT is 8
- The minimum value of MAXOUT is 8
- The maximum value of MAXOUT is 127

Token-Ring and Frame-Relay TESP Logical Stations: Token-Ring and frame-relay TESP logical stations might *delay* sending an acknowledgement for a certain number of frames received for a short period of time. For more information about delaying acknowledgement with Token-Ring and frame-relay TESP logical stations, see the description of the T2TIMER operand on the BUILD and LINE definition statements.

Note: Token-Ring and frame-relay TESP logical stations are always supporting a point-to-point link between adjacent Type 4 nodes when they are defined as logical *nonswitched* lines.

Token-Ring and frame-relay TESP logical stations always use a data link control which operates in modulo-128 mode and the following values apply:

- The default value of MAXOUT is 8
- The minimum value of MAXOUT is 1
- The maximum value of MAXOUT is 127

Link Stations in Adjacent Type 1 and Type 2.0 Nodes: The *coded* MAXOUT value is the *operational* MAXOUT value. The NCP link station is always the primary link station.

Link Stations in Adjacent Type 2.1 Nodes: In addition to the *coded* MAXOUT value, a value for MAXOUT is also supplied in the format-3 XID exchange. The *operational* MAXOUT value is the smaller of the following values:

- The value supplied in the format-3 XID exchange
- The *coded* MAXOUT value.

Therefore the *operational* MAXOUT value might be less than the *coded* MAXOUT value.

The *operational* MAXOUT value defines a limit for *sending* frames from NCP to the adjacent Type 2.1 link station. The *operational* MAXOUT value additionally defines a limit for *receiving* frames from the adjacent Type 2.1 link station.

If you want the MAXOUT value supplied in the format-3 XID exchanges to be the *operational* MAXOUT value, code MAXOUT=127 and MODULO=128.

Link Stations in Adjacent Type 4 Nodes:

Point-to-point connections: When two adjacent link stations in Type 4 nodes make contact and use the definition statement structure applicable only to *point-to-point* connections, a format-2 XID exchange takes place in which MAXOUT values are exchanged.

The primary station on the link uses the value of MAXOUT received in the *final* format-2 XID as the *operational* MAXOUT value. This MAXOUT value is the value coded on the relevant SDLCST definition statement with MODE=SECONDARY (or MODE=SEC) specified in the adjacent NCP.

The secondary station on the link uses the value of MAXOUT received in the *only* format-2 XID as the *operational* MAXOUT value. This MAXOUT value is the value coded on the relevant PU definition statement in the adjacent NCP.

When the values of MAXOUT have been exchanged, if one link station has received a value between 8 and 127 and one link station has received a value between 1 and 7, the *operational* MAXOUT value will be 7 for the link station that received a value for MAXOUT between 8 and 127.

Note: Do not code MAXOUT on the SDLCST definition statement with MODE=PRIMARY (or MODE=PRI) specified.

For clarity in the definition statements, the MAXOUT values coded on PU and SDLCST definition statements should be consistent, if possible.

Note: Primary and secondary roles may be determined based on the necessarily different subareas of the NCPs which are establishing contact or these roles may be predefined.

Multipoint connections: When two adjacent link stations in Type 4 nodes make contact and use the definition statement structure applicable only to *multipoint* connections, a format-2 XID exchange takes place in which MAXOUT values are exchanged.

The primary station uses the value of MAXOUT received in the format-2 XID as the *operational* MAXOUT value. This MAXOUT value is the value coded on the relevant PU definition statement specified in the adjacent NCP.

The secondary station on the link uses the value of MAXOUT received in the format-2 XID as the *operational* MAXOUT value. This MAXOUT value is that coded on the relevant PU definition statement in the adjacent NCP.

When the values of MAXOUT have been exchanged, if one link station has received a value between 8 and 127 and one link station has received a value between 0 and 7, the *operational* MAXOUT value will be 7 for the link station that received a MAXOUT value between 8 and 127.

Switched SDLC Link Stations: Do not code MAXOUT on a PU definition statement when DIAL=YES is coded on the GROUP definition statement. When the link station in the adjacent node is connected using switched procedures, VTAM passes the *coded* MAXOUT value to NCP when a switched connection is established between NCP and the adjacent link station. MAXOUT is defined on a PU definition statement, similar to the NCP PU definition statement, in a VTAM switched major node, whether defined or created dynamically. Except for in the case of switched subarea connections, the considerations of MAXOUT as defined on a PU definition statement for a nonswitched line apply to the MAXOUT value passed by VTAM.

For switched subarea connections, the values of MAXOUT which are exchanged and are used to derive the *operational* MAXOUT values are those coded in the VTAM switched major node rather than any values of MAXOUT present in the relevant SDLCST definition statements.

Frame-Relay and NTRI Frame Handler Switching: The *coded* MAXOUT value is used to calculate the maximum number of NCP buffers containing frames that NCP is to queue to the outbound physical unit for this frame relay frame handler subport or NTRI logical FHSP. The calculated maximum is the number of buffers required to hold MAXOUT frames of the maximum size as represented by the MAXFRAME operand of the LINE definition statement plus an additional MAXOUT buffers.

The following values apply:

- The default value of MAXOUT is 8
- The minimum value of MAXOUT is 4
- The maximum value of MAXOUT is 20

MAXOUT defines the congestion threshold used to manipulate the backward and forward explicit congestion indicators in frame-relay transmission frames. MAXOUT also defines when to discard frames. MAXOUT is not valid on the link-station subport physical unit.

METRIC

```
►►──METRIC=value──►◄
```

Use: Frame-relay IP resources, channel IP resources, and internal NCP to
                Model 900 point-to-point IP connections
Default: 1
Minimum: 1
Maximum: 15

Specifies a value used by NCPROUTE to evaluate new route information received
in Routing Information Protocol (RIP) packets over the interface defined by the PU
definition statement.

MODULO

```
        ┌─MODULO=8──────┐
►►──────┤               ├──►◄
        └─MODULO=──┬─8───┘
                   └─128─┘
```
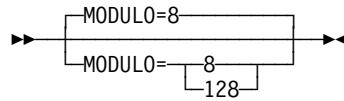
Use: Nonswitched link stations
Default: 8

Limits the maximum *possible* number of information frames (I-frames) that can be
sent to an adjacent link station without having been acknowledged, in other words,
an acknowledgement for the frames which have been sent is expected but has not
yet been received.

The *operational* maximum is described under the discussion associated with
MAXOUT. MODULO defines the maximum in terms of the data link control protocol.

If you code MODULO=8 or you do not code MODULO, the maximum value that
can be specified for MAXOUT is 7.

When you code MODULO=128, the maximum value that can be specified for
MAXOUT is 127.

When the *operational* MAXOUT value is determined to be 7 or less, the adjacent
link stations operates in modulo-8 mode.

If the *operational* MAXOUT value is determined to be 8 or more, the adjacent link
stations operates in modulo-128 mode.

Note: Not coding MODULO=128 causes the SDLC link between adjacent link
stations to operate in modulo-8 mode because the *operational* MAXOUT value
cannot be greater than 7. However, the reverse, namely coding MODULO=128, will
not necessarily cause the SDLC link between adjacent link stations to operate in
modulo-128 mode unless the manner in which the *operational* MAXOUT value is
determined is also taken into account.

Token-Ring and Frame-Relay Link Stations: They always operate in
        modulo-128 mode; MODULO=128 *must* be specified to be able to specify a
        value for MAXOUT greater than 7.

MONLINK

```
►►──┬────────────────────────────┬──►◄
    └─MONLINK=─┬─YES─┬────────────┘
              └─NO──┘
```
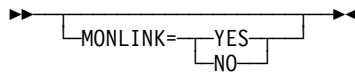
Use: ESCON logical link stations
Default: See text

Specifies whether the SMMF is to activate this ESCON logical link station. The value of MONLINK on the LINE definition statement determines how SMMF monitors the logical link station:

- If you code MONLINK=YES on the LINE definition statement and MONLINK=YES on the PU definition statement, SMMF activates the station when NCP is not owned by an external SSCP.

- If you code MONLINK=CONTINUOUS on the LINE definition statement and MONLINK=YES on the PU definition statement, SMMF continuously monitors the link station. If an external SSCP becomes an owner of NCP, the station will not be released.

If you code MONLINK=NO on the LINE definition statement or on the PU definition statement, SMMF does not activate the link station.

MONLINK is valid only when you code PUTYPE=5 on this PU definition statement.

If you code MONLINK=NO on the LINE definition statement for this logical link, the default is MONLINK=NO, and MONLINK=YES is not valid. Otherwise, the default is MONLINK=YES.

To load or dump NCP over an ESCON link, do the following:

- Code MONLINK=YES or MONLINK=CONTINUOUS on the LINE definition statement for the logical line.

- Code MONLINK=YES on the PU definition statement for the logical link station you want to use.

- Code XMONLNK=YES on the PU definition statement for the physical link station.

NETID

```
►►──┬──────────────┬──►◄
    └─NETID=name─┘
```
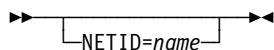
Use: Gateway NCP
Default: None

Specifies the name of the network that contains the adjacent link station you are defining. Use this keyword only for type 4 physical units (LNCTL=SDLC) or type 5 physical units (LNCTL=CA). *name* corresponds to NETID on the BUILD definition statement (native network) or the NETWORK definition statement (non-native network).

If you define channels, specify the name of the network that contains the adjacent VTAM represented by the link station you define. Use NETID if you define a link station (PUTYPE=5 is coded). If you do not specify this keyword, the adjacent VTAM is not in the native network.

NETID is not valid for NTRI frame handler logical lines.

If one gateway NCP is connected to another gateway NCP, the names specified for the NETID keywords on the PU definition statements for each gateway NCP must be the same.

If you code DIAL=YES and PUTYPE=4 for a switched subarea support group, the NETID values for all physical units in the group must be the same unless you code NETID=(*name*,ANY) on the GROUP definition statement.

For non-native network attachment support, if you code LNCTL=SDLC, PUTYPE=2, and XID=YES, or code LNCTL=CA and PUTYPE=2, you may code NETID as a VTAM-only keyword. Its value will be used by VTAM to verify the network identifier in the format-3 XID sent from the adjacent type 2.1 node. The value of NETID is not verified by NDF, and it need not match any of the values coded for the NETID keywords on the BUILD and NETWORK definition statements.

NETWORK

```
                  ┌─NETWORK=SNA──────┐
►►────────────────┼──────────────────┼────►◄
                  └─NETWORK=──┬─SNA───┬──┘
                             ├─IP────┤
                             └─FRELAY─┘
```
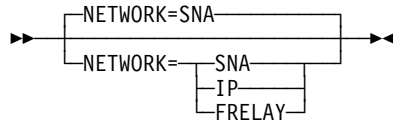
Use: Token-Ring physical lines
Default: SNA

Specifies the type of network or protocol this physical unit represents to the Token-Ring physical line.

NTRI Physical Lines: If you code NETWORK=IP, also code INTFACE on the LINE definition statement. If you code NETWORK=FRELAY, define at least one logical line with (ECLTYPE=(LOGICAL,FRELAY)).

3746 Model 900 Token-Ring Lines: If NETWORK=IP, then the PU statement must also include either

- The INTFACE keyword with an *interface name* suboperand, or
- The LADDR and P2PDEST keywords.

NETWORK=FRELAY is not valid.

NEXT
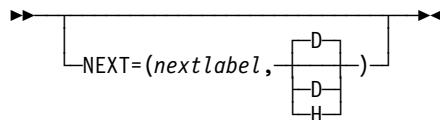
```
►►──────────────────────────────────────►◄
    │                       ┌─D─┐        │
    └─NEXT=(nextlabel,──────┼───┼──)─────┘
                            ├─D─┤
                            └─H─┘
```

Specifies a convention for constructing unique labels for the definition statements to be copied when you code the AUTOCOPY keyword on the LINE definition statement. NEXT is valid only when you code AUTOCOPY on the previous LINE defi-

nition statement. NEXT is required when you code AUTOCOPY on the LINE definition statement.

See AUTOCOPY on the LINE definition statement on page 2-196 for restrictions. See the *NCP, SSP, and EP Resource Definition Guide* for more information about using NEXT.

nextlabel

Default: None

Specifies the label for the first duplicated copy of this PU definition statement. *nextlabel* is required when you code AUTOCOPY on the LINE definition statement. The label consists of 1 to 8 characters; the first character must be an alphabetical character from A to Z, #, or @. NDF uses this *nextlabel* to determine what positions in the label to use for incrementing label names and the amount of the increment. The value you enter for *nextlabel* must be the same length as the statement label, and must differ from the label in such a way that NDF can identify a numeric field to use for incrementing. You can use only one field for incrementing. This field can increase in length as it is incremented, but it cannot increase so much that the total label length becomes greater than 8 characters.

The format of the statement *label* is

`<p>lo_num_field<s>`

The format of *nextlabel* is

`<p>next_num_field<s>`

, where:

- `label` is the name specified on this PU definition statement

- `nextlabel` is the label given to the first copy of this PU definition statement

- `p` is the prefix of the numeric field and must be the same for both labels

- `s` is the suffix of the numeric field and must be the same for both labels

NDF subtracts `lo_num_field` from `next_num_field` to obtain the value it uses to increment the numeric field for the next label.

The numeric fields cannot differ in length, but must differ in value. `next_num_field` must be greater than `lo_num_field`.

If NDF cannot determine how to increment a label, it issues an error message.

D|H

Default: D

Specifies whether the increment and the field to increment are decimal (D) or hexadecimal (H). What you specify here does not need to match what you code for AUTOCOPY or for NEXT on other resource definition statements being replicated.

NPACOLL

```
►►──┬─────────────────────────────────────────┬──►◄
     └─NPACOLL=─┬─(YES─┬──────────────┬──)──┬──┘
                │      └─,EXTended─────┘    │
                └─NO───────────────────────┘
```

Default: See text

Specifies whether performance data for this physical unit is to be collected by the network performance analyzer.

If you code LNCTL=CA on the GROUP definition statement and PUTYPE=5 on the PU definition statement, NPACOLL specifies that the data collected is from channel adapter traffic.

NO|YES
Specifies whether performance data is to be collected. NPACOLL=YES is valid only if you code (1) NPA=YES on the BUILD definition statement, and (2) NPACOLL=YES on the LINE definition statement or LEVEL2 or VIRTUAL=YES on the GROUP definition statement.

If you code VIRTUAL=YES on the GROUP definition statement, the default for user-written line control is NPACOLL=NO.

If you code NPA=(YES,DR) or NPA=(YES,DRTP) on the BUILD definition statement and PUDR=YES on the PU definition statement, the value you specify for NPACOLL is ignored. NCP uses NPACOLL=YES. If you code NPACOLL=YES, NDF uses NPACOLL=(YES,EXTENDED).

EXTended
Indicates that performance data collected by the network performance analyzer is to be forwarded to NPM using extended format COLLECT request units (RUs).

If you code ECLTYPE=LOGICAL or FRELAY=PHYSICAL on the GROUP definition statement or if you are defining an ESCON link, and if you code NPACOLL=YES on this PU definition statement, NDF uses NPACOLL=(YES,EXTENDED).

3746 Model 900 Token Ring: For 3746 Model 900 Token-Ring logical stations and internal NCP to Model 900 point-to-point IP connections, if you code NPACOLL=(YES,EXTENDED) on the GROUP definition statement and you do not code NPACOLL on the PU definition statement, NDF adds NPACOLL=(YES,EXTENDED) on the PU definition statement.

Table 2-104 shows the effect on NPACOLL when you code NPA=YES on the BUILD definition statement and certain keywords on the GROUP, LINE, and PU definition statements.

Table 2-104 (Page 1 of 2). NDF Values for NPACOLL on the PU Definition Statement

When You Code This	The Effect on NPACOLL Is
LINE: NPACOLL=YES	The default is NPACOLL=YES.
GROUP: ECLTYPE=PHYSICAL LINE: ADAPTER=TIC1 or TIC2	The default is NPACOLL=NO, and the only valid value is NPACOLL=NO.

Table 2-104 (Page 2 of 2). NDF Values for NPACOLL on the PU Definition Statement

When You Code This	The Effect on NPACOLL Is
GROUP: ECLTYPE=PHYSICAL LINE: ADAPTER=TIC3	The default is NPACOLL=NO.
GROUP: ECLTYPE=LOGICAL	The default is NPACOLL=NO. (See description of EXTENDED.) (3746 Model 900 Token Ring: see note.)
GROUP: ETHERNET=PHYSICAL	The only valid value is NPACOLL=NO.
GROUP: FRELAY=PHYSICAL	The default is NPACOLL=NO. (See description of EXTENDED.)
GROUP: FRELAY=LOGICAL	NDF adds NPACOLL=NO. The only valid value is NPACOLL=NO.
PU: LMI=NO	The only valid value is NPACOLL=NO.
ESCON resources	The default is NPACOLL=(YES,EXTENDED). (See description of EXTENDED.)
3746 Model 900 SDLC resources	The default is NPACOLL=(YES,EXTENDED). (See description of EXTENDED.)
GROUP: ISDN	NDF adds NPACOLL=NO. The only valid value is NPACOLL=NO.

NPATP
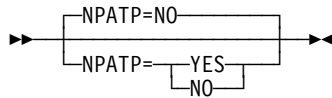
```
              ┌─NPATP=NO──────┐
►►────────────┴───────────────┴───────►◄
   └─NPATP=──┬─YES─┬──┘
             └─NO──┘
```

Use: SDLC resources
Default: NO

Specifies whether you want network performance data to be collected by PIU trans-
mission priority on the physical unit in addition to basic network performance data.
Code NPATP=YES if you want network performance data collected by transmission
priority. If you code NPATP=YES, you must also code NPACOLL=YES for this
resource.

NPATP is not valid if you code ECLTYPE, ETHERNET, FRELAY, LEVEL2, NCST,
or VIRTUAL=YES on the GROUP definition statement.

PASSLIM

```
►►─────┬────────────────────┬──────►◄
       └─PASSLIM=count─┘
```
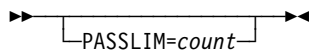
Use: Nonswitched link stations
Default: See text
Minimum: 1
Maximum: 254

Specifies the maximum number of units of data that NCP can send at one time
when the resource represented by the PU definition statement is sharing the
resource represented by the associated LINE definition statement.

SDLC Stations: Specifies the maximum number of consecutive path information units (PIUs) that NCP can send to an adjacent link station when that link station is due to receive PIUs.

It is important to distinguish between the value of PASSLIM as specified on the PU definition statement, the *coded* PASSLIM value, and the value of PASSLIM used in operation of the link, the *operational* PASSLIM value. If a value for PASSLIM is not specified on the PU definition statement, the *coded* PASSLIM value defaults.

When NCP is a primary link station, NCP will service a queue of PIUs for a secondary link station. Queued PIUs are sent to the secondary link station until a limit defined by the current size of the queue, the *operational* PASSLIM value or the *operational* MAXOUT value is reached, whichever of these limits applies first.

The one or more secondary link stations are selected for outbound service according to a static service order table (SOT) when LSPRI on the associated LINE definition statement is not LSPRI=LINK, or by dynamic service order queues (SOQs) when LSPRI=LINK.

If NCP and the one or more secondary link stations are operating *half-duplex* and HDXSP=NO is specified on the LINE definition statement, scheduling of the one or more link stations will be from queues defined by the SOT since LSPRI=LINK is invalid. Receiving PIUs from the secondary station is scheduled whenever the sending of PIUs reaches a limit, assuming the delay defined by the PAUSE operand of the associated LINE definition statement is not in effect.

In the case of a *half-duplex* line with HDXSP=NO, PASSLIM can define how the line resource is shared between the primary link station and the one or more secondary link stations.

If NCP and the one or more secondary link stations are operating *half-duplex* and HDXSP=YES on the LINE definition statement, scheduling of the one or more link stations might be from queues defined by the SOT or from SOQs, depending on the specification of the LSPRI operand. To receive PIUs from a secondary station, the primary station polls one of the secondary stations only when the following conditions exist:

- The primary station does not have any PIUs to send

- The limit defined by the *operational* MAXOUT value has been reached for *all* secondary stations managed by the primary.

If NCP and the one or more secondary link stations are operating *full-duplex*, scheduling of the one or more link stations can be from queues defined by the SOT or from SOQs, depending on the specification of the LSPRI operand. Except for the sending of a frame containing the *poll* indicator, which can take place following the sending of any PIU, receiving PIUs from the secondary station is scheduled independently of sending PIUs.

In the case of a *half-duplex* line with HDXSP=YES or a *full-duplex* line, you can use PASSLIM to define how the line resource is shared between the secondary link stations. If there is only one secondary link station,

PASSLIM has no appreciable significance and may be defined as the maximum, 254.

Attention: Receiving PIUs from the secondary station is scheduled independently of sending PIUs but sending PIUs to the secondary station may not be scheduled independently of receiving PIUs. If there are multiple secondary link stations, the *operational* PASSLIM value should not significantly exceed the *operational* MAXOUT value. Otherwise, the output to *any* of the secondary link stations can be suspended pending completion of sending PIUs, up to the PASSLIM limit, to one particular secondary link station. This happens when the secondary link station is also scheduled to be polled to have the frames, sent during earlier outbound service, acknowledged. Inbound service is always in the sequence defined by the SOT.

When NCP is a secondary link station, queued PIUs are sent up to the limit defined by the current size of the queue or the *operational* MAXOUT value, whichever limit applies first. Queued PIUs are sent after a frame containing a *poll* indicator is received and, once one of the limits is reached, NCP sends a frame containing the *final* indicator.

3746 Model 900: When NCP is a secondary link station, in addition to the limit imposed by the current size of the queue and the *operational* MAXOUT value, queued PIUs are sent up to the limit defined by the *operational* PASSLIM value, whichever limit applies first.

PASSLIM defaults to:

- 1 for a link station in a Type 1 or 2.0 node

- 1 for a link station in a Type 2.1 node where NCP has the primary role

- 254 for a link station in a Type 2.1 node where NCP has the secondary role

- 254 for a link station in a Type 4 node

Link Stations in Adjacent Type 1 and Type 2.0 Nodes: The *operational* PASSLIM value is the same as the *coded* PASSLIM value. NCP is always the primary link station.

Link Stations in Adjacent Type 2.1 Nodes: If the *operational* MAXOUT value is the *coded* MAXOUT value, the *operational* PASSLIM value is the *coded* PASSLIM value. If the *operational* MAXOUT value is the value corresponding to MAXOUT as received from the adjacent link station in the format-3 XID exchange, the *operational* PASSLIM value is the smaller of the following values:

- The *coded* PASSLIM value
- The *operational* MAXOUT value

Link Stations in Adjacent Type 4 Nodes:

Point-to-point connections: When two adjacent link stations in Type 4 nodes make contact and they use the definition statement structure applicable only to *point-to-point* connections, the *operational* PASSLIM values depend on which link station will be the primary and which the secondary.

The primary station on the link uses the value of PASSLIM that is coded on the relevant SDLCST definition statement with MODE=PRIMARY (or MODE=PRI).

The secondary station on the link uses the value of PASSLIM that is coded on the relevant SDLCST definition statement with MODE=SECONDARY (or MODE=SEC).

Notes:

1. The *coded* PASSLIM values on the PU definition statements are never used.

2. Primary and secondary roles can be determined based on the necessarily different subareas of the NCPs which are establishing contact or these roles may be predefined.

Multipoint connections: When two adjacent link stations in Type 4 nodes make contact and they use the definition statement structure applicable only to *multipoint* connections, the *operational* PASSLIM value is the *coded* PASSLIM value as specified on the PU definition statements.

Switched SDLC Link Stations: Do not code PASSLIM on a PU definition statement when DIAL=YES is coded on the GROUP definition statement. When the peripheral link station in the adjacent node is connected using switched procedures, VTAM passes the *coded* PASSLIM value to NCP when a switched connection is established between NCP and the adjacent link station. PASSLIM is defined on a PU definition statement, similar to the NCP PU definition statement, in a VTAM switched major node, whether defined or created dynamically. All the considerations of PASSLIM as defined on a PU definition statement for a nonswitched line apply to the value of PASSLIM passed by VTAM.

For switched subarea connections, PASSLIM is not coded in the VTAM switched major node and PASSLIM is determined as described for nonswitched subarea connections.

NTRI Physical Resources (For NCP V7R1 and NCP V7R2): PASSLIM controls sharing of an NCP Token-Ring adapter for outbound traffic when the adapter is used for both SNA and IP protocols.

PASSLIM specifies a limit for how many consecutive frames may be transmitted for the network protocol defined by NETWORK on this PU definition statement. Service for outbound traffic then switches to the network protocol defined for the other protocol if a PU definition statement for the other protocol is defined. Otherwise, outbound traffic is not interrupted. For more information, see *NCP, SSP, and EP Resource Definition Guide*.

The default value of PASSLIM is 1. An NTRI physical resource is represented by a link station in a Type 1 node.

NTRI Physical Resources (for NCP V7R3 and later): A mechanism for resource sharing between network protocols associated with the use of COMRATE on the PU definition statement replaces the mechanism associated with PASSLIM. See COMRATE on the PU definition statement on page 2-375 for more information.

PU

PIUCHECK
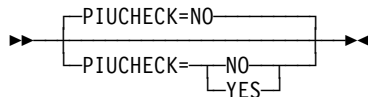
```
              ┌─PIUCHECK=NO──────┐
►►─┬──────────────────────────┬─►◄
   └─PIUCHECK=─┬─NO─┐─────────┘
              └─YES─┘
```

Use: Frame-relay logical subarea stations, ISDN logical subarea stations,
               SDLC subarea stations, and Token-Ring logical subarea stations
Default: NO

Specifies whether a checksum validation is performed on PIUs sent and received
by this station. For the checksum support to be enabled, the adjacent link station
must also have PIUCHECK=YES coded.

PROTOCOL
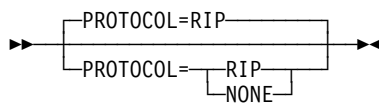
```
              ┌─PROTOCOL=RIP─────┐
►►─┬──────────────────────────┬─►◄
   └─PROTOCOL=─┬─RIP──┐────────┘
              └─NONE─┘
```

Use: Frame-relay IP resources, channel IP resources, and internal NCP to
               Model 900 point-to-point IP connections
Default: RIP

Specifies that RIP can manage information about the router interface specified on
the PU definition statement. NCP uses the NCPROUTE program and RIP to
transmit information about this interface to other routers. Code
PROTOCOL=NONE when you define interfaces that use other protocols.

Note: RIP Version 2 is supported starting with NCP V7R6. The version of RIP to
be used is configured in the NCPROUTE program provided by IBM TCP/IP.

PUCB

```
►►─┬──────────────┬─►◄
   └─PUCB=name─┘
```
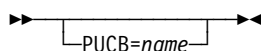
Use: User-written code and IP
Default: None

Specifies the name of a user-defined control block associated with a virtual physical
unit. PUCB is valid only if you code VIRTUAL=YES on the GROUP definition
statement.

Internet Protocol: When NCST=IP is specified on the GROUP definition state-
               ment and you do not code PUCB on an associated PU definition statement,
               NDF adds PUCB with an appropriate value to the generation definition.

PUDR

```
►►─┬──────────────┬─►◄
   └─PUDR=─┬─NO──┐─┘
          └─YES─┘
```
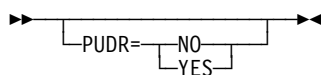
Use: Dynamic reconfiguration on a nonswitched link
Default: See text

Specifies whether the physical unit can be deleted from the network using the dynamic reconfiguration function. PUDR is not valid if dynamic reconfiguration is not supported, the physical unit is a type 4, or the physical unit is on a switched link.

You must specify PUDR=YES to use the MOVE PU function to delete resources from one line and move them to another line.

For casual connection, code PUDR=NO when MODE=SECONDARY on the GROUP definition statement.

Specify PUDR=NO if you do not want this physical unit deleted.

If dynamic reconfiguration is supported, the physical unit is not a type 4 physical unit, and the physical unit is on a nonswitched line, the default is YES.

ESCON Links: The PUDR keyword is valid only for a peripheral logical station; do not code PUDR for physical stations. The default is YES when dynamic reconfiguration is supported and PUTYPE=2.

Table 2-105 shows the effect on PUDR when you code certain keywords on the GROUP definition statement.

Table 2-105. NDF Values for PUDR on the PU Definition Statement

When You Code This	The Effect on PUDR Is
ECLTYPE=(PHYSICAL|LOGICAL, PERIPHERAL|SUBAREA|ANY)	NDF adds PUDR=NO. The only valid value is PUDR=NO.
ETHERNET=PHYSICAL	The default is PUDR=NO. The only valid value is PUDR=NO.
ECLTYPE=(LOGICAL,FRELAY) or FRELAY=PHYSICAL (link-station subport)	NDF adds PUDR=NO. The only valid value is PUDR=NO.
ECLTYPE=(LOGICAL,FRELAY) or FRELAY=PHYSICAL (FHSP) and PUDRPOOL definition statement or DYNPOOL value on BUILD > 0	The default is PUDR=YES.
FRELAY=PHYSICAL (frame-relay IP PU on a leased physical line)	The default is PUDR=YES.
FRELAY=LOGICAL	The default is PUDR=NO. The only valid value is PUDR=NO.
FRELAY=(LOGICAL,IP) (frame-relay IP PU on a leased logical line)	The default is PUDR=YES. (However, IP PU in first position must be PUDR=NO.)

PUFVT
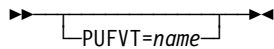
```
►►─┬───────────────┬─►◄
   └─PUFVT=name─┘
```

Use: User-written code
Default: None

Specifies the name of the FVT associated with a virtual physical unit. PUFVT is required if you code VIRTUAL=YES on the GROUP definition statement. This keyword is not valid if you specify VIRTUAL=NO.

An EXTRN statement for each unique FVT symbol must be included in the user source code that is copied into the tables during assembly. For more information, see SRCHI on the GENEND definition statement on page 2-93.

The function vector table must be preassembled and included in the user object modules. For more information, see INCHI on the GENEND definition statement on page 2-96.

Internet Protocol: When NCST=IP is specified on the GROUP definition statement, only PUFVT=CXSXFVT is valid. If you do not code PUFVT, NDF adds PUFVT=CXSXFVT to the generation definition.

PUNTFY

```
►►─┬──────────────┬─►◄
   └─PUNTFY=─┬─1─┬─┘
            ├─2─┤
            └─3─┘
```
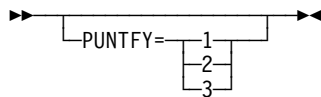
Use: User-written code and IP
Default: See text

Specifies the use of the notify byte in user-written control blocks and whether control is passed to the user-written notify task.

PUNTFY=1 sets and resets the virtual route blocked bit in the notify byte of the NPB control block and passes control to the notify task. PUNTFY=2 sets and resets the virtual route blocked bit but does not pass control to the notify task. PUNTFY=3 sets and resets the virtual route blocked bit in both the programmed resource virtual line block (VLB) and programmed resource logical unit block (NPB) control blocks and does not pass control to the notify task.

PUNTFY is valid only if you specify VIRTUAL=YES on the GROUP definition statement. If you code VIRTUAL=YES and omit PUNTFY, the default is 1.

Internet Protocol: When NCST=IP is coded on the GROUP definition statement, the default value of PUNTFY is 1.

PUTYPE

```
►►──────────────────────────────►◄
    └─PUTYPE=─┬─1────┬─
             ├─2────┤
             ├─(1,2)┤
             ├─4────┤
             └─5────┘
```
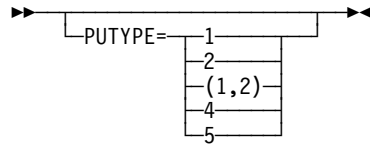
Default: See text

Specifies the physical unit type of the SDLC device you are defining. For a group of switched SDLC links, PUTYPE=(1, 2) specifies that physical units of either type 1 or type 2 can communicate with the communication controller over the same link.

PUTYPE=1 and PUTYPE=2 are valid for links that use either nonswitched or switched line control procedures (DIAL=NO or DIAL=YES on the GROUP definition statement).

PUTYPE=2 represents either a type 2 physical unit or a type 2.1 physical unit.

If you code LNCTL=CA on GROUP, you must define either PUTYPE=2 or PUTYPE=5 or PUTYPE=1 for each line under the GROUP definition statement. PUTYPE=2 defines any channel adapter that provides the peripheral channel attachments; PUTYPE=5 defines any channel adapter that provides the subarea channel attachments; PUTYPE=1 defines any channel adapter that provides the internet protocol channel attachments. PUTYPE=5 is the default.

To define a switched subarea support line, specify PUTYPE=4 and DIAL=YES on the GROUP definition statement.

You may have more than one PUTYPE=4 for an ES/9370 or 4300 processor if you define MODE=PRIMARY and do not code SDLCST on the LINE definition statement for primary multipoint links.

PUTYPE=4 represents a communication controller containing an NCP (adjacent subarea) with which this NCP can communicate. Code one PU definition statement with PUTYPE=4 specified for each subarea link following the LINE definition statement for the link. You do not need to code a SERVICE definition statement for this type of link.

Specify PUTYPE=4 for an NCP attached by an SDLC line through an integrated communication adapter (ICA) to an ES/9370 or 4300 processor.

PUTYPE=4 is not valid if you specify both of the following:
- VIRTUAL=YES on the GROUP definition statement for this physical unit
- CHNLZ=YES on the LINE definition statement

ESCON Links: Specify PUTYPE=1 to define the link station for the physical ESCON connection. Specify PUTYPE=2 for physical units on logical peripheral lines or PUTYPE=5 for physical units on logical subarea lines or PUTYPE=1 for physical units on logical internet protocol lines.

APPN: You need to define type 2.1 physical units to support APPN functions for NCP. Code PUTYPE=2 and XID=YES.

Spare or Redefinable Lines: PUTYPE=4 and PUTYPE=5 are not valid.

Table 2-106 on page 2-412 shows the effect on PUTYPE when you code certain keywords on the GROUP definition statement.

Table 2-106. NDF Values for PUTYPE on the PU Definition Statement

When You Code This on GROUP	The Effect on PUTYPE Is
DIAL=YES	The default is PUTYPE=(1,2).
DIAL=NO or CHANLNK=YES	The default is PUTYPE=2.
LNCTL=CA	The default is PUTYPE=5.
ESCON physical link station	PUTYPE=1 is required.
Parallel transmission groups	The default is PUTYPE=5.
ECLTYPE=PHYSICAL or ECLTYPE=(LOGICAL,FRELAY)	NDF adds PUTYPE=1. The only valid value is PUTYPE=1.
ECLTYPE=(LOGICAL,SUBAREA)	NDF adds PUTYPE=4. The only valid value is PUTYPE=4.
ECLTYPE=(LOGICAL,PERIPHERAL)	NDF adds PUTYPE=2. The only valid value is PUTYPE=2.
NCST=IP	NDF adds PUTYPE=2. The only valid value is PUTYPE=2.
ETHERNET=PHYSICAL	NDF adds PUTYPE=1. The only valid value is PUTYPE=1.
FRELAY=PHYSICAL	NDF adds PUTYPE=1. The only valid value is PUTYPE=1.
FRELAY=LOGICAL or FRELAY=(LOGICAL,SUBAREA)	NDF adds PUTYPE=4. The only valid value is PUTYPE=4.
FRELAY=(LOGICAL,PERIPHERAL)	NDF adds PUTYPE=2. The only valid value is PUTYPE=2.
ISDN=PHYSICAL ISDN=(LOGICAL,B)	NDF adds PUTYPE=1. The only valid value is PUTYPE=1.
ISDN=(LOGICAL,SUBAREA)	NDF adds PUTYPE=4. The only valid value is PUTYPE=4.
ISDN=(LOGICAL,PERIPHERAL)	NDF adds PUTYPE=2. The only valid value is PUTYPE=2.

P2PDEST

```
►►──┬──────────────────────────────────┬──►◄
    └─P2PDEST=(──address──,──┬──D──┬──)─┘
                            ├──H──┤
                            └──DD─┘
```
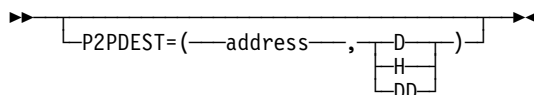
Use: Frame-relay IP connections, channel IP connections, and internal
            NCP to Model 900 point-to-point IP connections

Default: None

Specifies the destination internet address for the associated internet host defined by this PU definition statement.

Frame-Relay IP Connections: If P2PDEST is not coded, then Inverse ARP processing is initiated to resolve the destination internet address for the associated frame-relay internet host. If it is coded, *address* is associated with the hardware address specified on ADDR or DLCI.

Channel IP Connections: P2PDEST is required when specifying channel internet resources. If you code LADDR on this PU definition statement, code P2PDEST as well. If you code INTFACE on this PU definition statement, code P2PDEST on the associated IPLOCAL definition statement.

Internal NCP to Model 900 Point-to-Point IP Connections: The P2PDEST keyword is required for an internal NCP to Model 900 point-to-point IP connection and must be coded either on the PU statement or on the associated IPLOCAL statement, as follows:

- On the PU statement that defines the connection, or
- On the IPLOCAL statement that is referenced (via the *name* suboperand of the INTFACE keyword) by the PU statement that defines the connection

The 3746 Model 900 Controller Configuration and Management (CCM) definitions will also include the IP address for the 3746 Model 900 side of the internal NCP to Model 900 point-to-point IP connection. This CCM-defined address must match the corresponding NCP P2PDEST definition.

address is a 4-byte internet address that may be specified in either decimal, hexadecimal, or dotted decimal format.

To code the address as a decimal number, use the format P2PDEST=(*dddddddddd*,D), where D means decimal representation. For example, code the address 128.255.16.1 as P2PDEST=(2164199425,D).

Note: The value 2147483648 (2^{31}) is not valid.

To code the address as a hexadecimal number, use the format P2PDEST=*hhhhhhhh*. For example, code the address 128.255.16.1 as P2PDEST=80FF1001.

To code the address as a dotted decimal number, use the format P2PDEST=*n0.n1.n2.n3*, where *n0–n3* are numbers between 0 and 255. For example, P2PDEST=128.255.16.1.

See the *NCP, SSP, and EP Resource Definition Guide* for more information about internet addresses.

REMOTTO
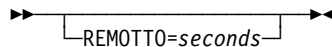
```
►►──┬────────────────────┬──►◄
    └─REMOTTO=seconds─────┘
```

Use: Token-Ring logical subarea resources
Default: Value of REMOTTO on the BUILD definition statement.
Minimum: 0.6
Maximum: 20.0

Specifies the Token-Ring reply timer (T1) value for stations connected to a remote Token Ring. This keyword is valid only if ECLTYPE=(LOGICAL,SUBAREA).

REMOTTO on LINE does not apply to NTRI frame handler logical lines. The value specified on the BUILD statement or its default will be used during test resolve processing for NTRI frame handler logical line connections.

Note: Timer values for logical peripheral lines may be specified on the BUILD definition statement.

RETRIES

```
►►──┬──────────────────────────────┬──►◄
    └─RETRIES=(,──┬───┬──┬─────┬──)─┘
                  └─t─┘  └─,n──┘
```
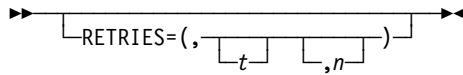
Use: Nonswitched links
Default: None
Minimum: None
Maximum: None
Example: RETRIES=(,255,127)

Specifies, together with RETRIES on the LINE definition statement, the number of attempts to be made to recover from errors occurring during transmission to or from the physical unit represented by this PU definition statement. See the description of RETRIES on the LINE definition statement for a description of the *t* and *n* suboperands.

For casual connection, RETRIES is not valid for secondary physical units.

To prevent NCP from losing contact with an adjacent SDLC-link-attached communication controller while the adjacent controller is accessing its disk during a load or dump over the link, do the following:

1. Specify REPLYTO and RETRIES on the GROUP, LINE, or PU definition statement that defines the attachment to the adjacent controller such that the retry time is at least 3 minutes.

2. Set the link activity timer on the MOSS LKP function Common Options panel of the adjacent controller to a value equal to or greater than both of the following values:

   * REPLYTO on the GROUP definition statement that defines the attachment to the adjacent controller

   * The *t* suboperand of RETRIES on the GROUP, LINE, or PU definition statement that defines the attachment to the adjacent controller

ROLE

```
         ┌─ROLE=NEG────────┐
►►──┬────┴─────────────────┴──┬──►◄
    └─ROLE=──┬─NEG─┬──────────┘
             └─PRI─┘
```
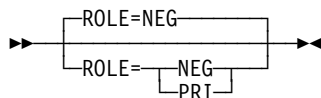
Use: Switched SDLC lines
Default: NEG

Specifies the data link control role which the station will assume at CONTACT time if format-3 XIDs were exchanged during the connect sequence.

You can code ROLE as primary (PRI), which means that the station controls the session, or as negotiable (NEG), which means that the station can assume either the primary or secondary role.

Switched Frame-Relay Physical Lines: ROLE=PRI is the only valid value.

SEGMENT

```
►►──┬────────────────────┬──►◄
    └─SEGMENT=─┬─1─┬──────┘
              └─2─┘
```
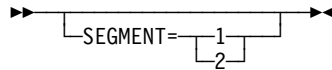
Default: See text

Specifies which link segment a station is on and allows NCP to set up LPDA2 tests accordingly. SEGMENT is valid only if you specify LPDATS=LPDA2 on the LINE definition statement. SEGMENT=1 indicates the station is on segment number 1; SEGMENT=2 indicates the station is on segment number 2.

A value of 2 is valid only if you specify TAILING=YES on the LINE definition statement.

The default value is 2 if TAILING=YES and 1 if TAILING=NO.

SNETMASK

```
►►──┬──────────────────────────────┬──►◄
    └─SNETMASK=─address_mask─┘
```
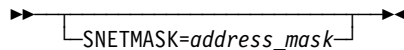
Use: Internet protocol subnetworks: Channel IP resources, frame-relay IP resources, and internal NCP to Model 900 point-to-point IP connections

Default: See text

Specifies a unique subnetwork mask. The subnetwork mask determines which portion of an internet address, defined by the LADDR keyword, identifies the subnetwork associated with this internet interface. The SNETMASK value must contain the network portion of the internet address defined by the LADDR keyword.

When VSUBNETS=NO on the BUILD statement:
> You can define a subnetwork mask for each network by coding SNETMASK on the first PU (or IPLOCAL) definition statement that defines an interface to that network. If you omit SNETMASK on subsequent PU (or IPLOCAL) definition statements for that network, the SNETMASK value coded on the first PU (or IPLOCAL) definition statement with a LADDR value for that network is the default.

When VSUBNETS=YES on the BUILD statement:
> You can define one or more subnet masks for each network by coding SNETMASK on one or more of the PU (or IPLOCAL) statements that define interfaces to that network. If you omit SNETMASK on any of the PU (or IPLOCAL) definition statements for that network, a default value will **not** be supplied, and that interface will not have subnetting.

Internal NCP to Model 900 Point-to-Point IP Connections: The 3746 Model 900 Controller Configuration and Management (CCM) definitions will also include the IP subnetwork mask for the 3746 Model 900 side of the internal NCP to Model 900 point-to-point IP connection. This CCM-defined address must match the corresponding NCP SNETMASK definition.

address_mask is a 4-byte address that must be specified in either dotted decimal or hexadecimal notation. All bits in the network portion of the SNETMASK must be on. For class A, B, and C networks, note the following conventions:

Class A   The first 8 bits must all be on (255.n1.n2.n3).
Class B   The first 16 bits must all be on (255.255.n2.n3).
Class C   The first 24 bits must all be on (255.255.255.n3).

The remainder of the SNETMASK value identifies the host and subnetwork
addresses where you set bits on to identify the subnetwork address portion and set
bits off to identify the host address portion.

To code the address as a dotted decimal number, use the format
SNETMASK=*n0.n1.n2.n3* where *n0–n3* are numbers from 0 to 255. For example,
code SNETMASK=255.255.240.0.

To code the address as a hexadecimal number, use the format
SNETMASK=*hhhhhhhh*. For example, code the address 255.255.240.0 as
SNETMASK=FFFFF000.

Note: Avoid defining a subnetwork mask that causes the host address to contain
all zeros or all ones. NCP treats datagrams with a resulting destination host
address of all zeros as a network address, and all ones as a broadcast address,
and discards them.

SPOLL

Use: Frame relay (link-station subport only)
              NTRI frame handler logical line (link-station subport only)
              ISDN (physical PU only)
Default: 6
Minimum: 1
Maximum: 255

Specifies the minimum frequency (number of T391 intervals) at which LMI Status
Enquiry Full messages are to be sent to the adjacent node. SPOLL is valid only
when the LMI keyword (not LMI=NO) is coded on this PU definition statement. For
more information, see TIMERS on the PU definition statement on page 2-421.

SRT
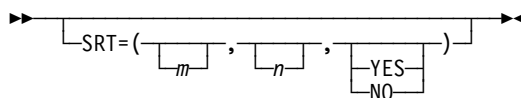
Default: See text
Example: SRT=(65280,32766,YES)

Specifies a threshold value for the total number of transmissions and for the total
number of error retries associated with this physical unit. When either of these
thresholds is reached, NCP generates a record maintenance statistics (RECMS) or
problem determination statistics (PDstats) network management vector table
(NMVT) PIU to inform the host of the condition.

m

   Default: 32768

Minimum: 1
Maximum: 65280

Specifies the threshold for total transmissions.

Attention: A transmission threshold value of 1 is not recommended because LPDA testing can cause other processing to be locked out.

n

Default: 32768
Minimum: 1
Maximum: 65280

Specifies the threshold for error retries.

YES│NO

Default: See text

Specifies whether LPDA tests for the total transmissions counter are to be performed. If you specify SRT=(*m*,*n*,YES), the LPDA tests are performed when the total transmissions counter reaches the threshold specified by *m*. This suboperand:

- Is not valid for ESCON links or 3746 Model 900 lines
- Is ignored for NTRI lines
- Has a default of NO for 3745 frame-relay lines
- Has a default of YES for SDLC lines

For IBM 786x modems extended LPDA2 capability lines connected to port B, C, or D, code large values for the SRT thresholds or allow SRT to default to the maximum value. This reduces the performance degradation of the port A line while LPDA2 tests are being run on the line on port A to generate statistical records for lines on ports B, C, and D. Code NO as the third suboperand of the SRT keyword to block the LPDA2 test when the total number of transmissions counter reaches its threshold.

SRT is not valid if you specify LEVEL5=USER on the GROUP definition statement. The thresholds specified by SRT do not apply to physical units added using dynamic reconfiguration. Dynamically added physical units use the default.

NTRI: You can specify reporting thresholds for NTRI logical lines. You can use the total transmit threshold *m* as the transmit counter and the receive counter. You can use the total retries counter *n* as the transmit error counter and the receive error counter. You can query and alter these thresholds using the NetView program. The third suboperand of the SRT keyword is ignored. The values you code for SRT do not affect NTRI processing.

NTRI Frame Handler Logical Links: The default is SRT=(65280,65280,YES).

3746 Model 900 Token Ring: The SRT keyword is not valid for 3746 Model 900 Token-Ring resources.

Frame Relay: The default is SRT=(65280,65280,NO). NO is the only valid value for the third suboperand because LPDA tests are not supported.

For frame-relay logical resources, you can specify reporting thresholds. Use the total transmit threshold m as the transmit counter threshold and the receive counter threshold. Use the total error retries counter n as the transmit and receive error counter threshold.

For frame-relay physical resources, SRT is valid only for the link-station subport physical unit, that is, the first PU definition statement under a LINE definition statement in a group of frame-relay physical lines. Use the total transmit threshold m as the transmission counter threshold. Use the total error retries counter n as the transmit and receive error counter threshold. These counters represent the entire frame-relay physical port, not just the link-station subport.

To query and alter these thresholds, use the NetView program.

3746 Model 900 Frame Relay: The SRT keyword is not valid for 3746 Model 900 frame-relay resources.

ISDN: The SRT keyword is not valid for ISDN resources.

ESCON Links: The first two suboperands are valid for both physical and logical stations. The third operand is not valid.

SSAP
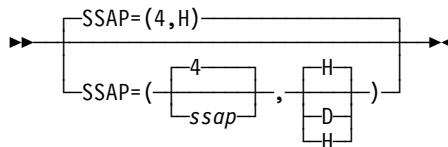
Use:	Token-Ring logical subarea PUs
Default:	(4,H)
Minimum:	(4,H) or (4,D)
Maximum:	(9C,H) or (156,D)

Specifies the source service access point (SSAP) to use for a subarea node connection. Use the SSAP keyword to provide addressing uniqueness for those Token-Ring logical subarea links that are associated with duplicate locally administered MAC addresses.

SSAP definition format and definition rules are as follows:

- SSAP=(*ssap*,D) (D means SSAP is specified in decimal)
- SSAP=(*ssap*,H) (H means SSAP is specified in hexadecimal)

—where *ssap* must be a multiple of 4. If no type is specified, hexadecimal is assumed.

TGCONF
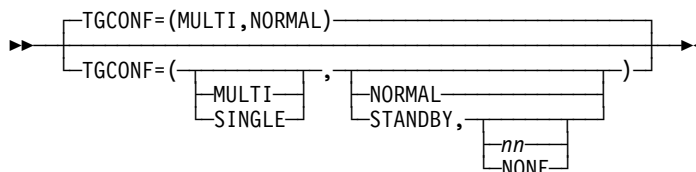
Use: Transmission group subarea link stations

Default: (MULTI,NORMAL)

Specifies whether this subarea link station is in a multilink or single-link transmission group. TGCONF is valid only if you specify PUTYPE=4.

| For switched PUs, TGCONF can be coded in the VTAM switched major node with
| the appropriate level of VTAM. Values coded for TGCONF in the VTAM switched
| major node will override the values coded or taken as default in the NCP definition.

Multilink Transmission Groups: Code TGCONF=MULTI or omit TGCONF for all subarea link stations in a multilink transmission group.

Single-Link Transmission Groups: You do not need to code TGCONF for a single-link transmission group. However, coding TGCONF=SINGLE for a transmission group that contains a single line can improve NCP performance for the transmission group.

Standby Stations: TGCONF=STANDBY designates if a subarea station is to be used in Hot-Standby mode. TGCONF=STANDBY is not valid when TGCONF=SINGLE is coded.

TGCONF=(MULTI,STANDBY,*nn*) designates that a subarea station is to be used for hot-standby and congestion relief.

"Hot standby" means that the station will be used to transmit data when all NORMAL stations in the MLTG are inoperative.

"Congestion relief" means that the subarea station transmits data to relieve congestion in the transmission group configuration if queued data reaches the value specified by *nn*. *nn* indicates the percent reached of a threshold specified by the *totthresh* suboperand of the ERn keyword on the associated PATH statement. The *totthresh* suboperand defines the total number of bytes that can be queued for transmission to an MLTG before NCP takes control measures to slow down the TG traffic. Valid values for *nn* are 3, 6, 12, 25, 50, 75, 87, 93, 97, and 100. NONE designates that the station is used only for hot-standby and not for congestion relief.

TGN

```
              ┌─TGN=1──────────┐
►►────────────┼────────────────┼────►◄
              └─TGN=──┬─number─┬┘
                      └─ANY────┘
```
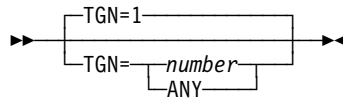
Default: 1

Minimum: 1

Maximum: 255

Specifies a transmission group number (TGN). TGN is valid only if you code PUTYPE=4 on this PU definition statement or if you code PUTYPE=5 on this PU definition statement and LNCTL=CA on the GROUP definition statement. (If you code PUTYPE=2 on this PU definition statement, TGN is also valid for APPN, but it becomes a VTAM-only keyword.)

If you define channels, TGN specifies the transmission group number of the subarea link associated with this physical unit. *number* is a specific transmission

group number; code this value for one of the ER*n* keywords on at least one PATH definition statement in this generation. The transmission group number is used during the contact procedure to associate the physical unit control blocks with the transmission group.

If you code ANY instead of a numeric value, the transmission group number is assigned during the contact procedure with an adjacent subarea. This group number is the value specified for TGN on the PU definition statement in the adjacent subarea. Therefore, either this PU definition statement or the PU definition statement in the adjacent subarea's NCP must have a nonzero transmission group number defined for TGN.

To establish two-way communication between adjacent NCPs, TGN for a particular link must be the same at both ends of the link. Therefore, be sure that the transmission group numbers for this NCP correspond to the numbers in all adjacent NCPs. ANY can be used at either end of the link to simplify coding, but not at both ends. Do not try to dynamically switch a link within a transmission group without deactivating that link.

A multilink transmission group can contain no more than 255 active physical and logical lines, so do not code the same value for TGN on more than 255 lines that may be active at the same time.

Be sure that the values for TRANSFR are identical for all the links within the same transmission group. Failure to do this will result in link activation problems.

TGN is not valid for switched subarea lines.

For parallel transmission group support for channel links, code TGN=*number* or TGN=ANY when LNCTL=CA and PUTYPE=5 on the GROUP definition statement.

You can assign more than one Token-Ring or frame-relay logical line to the same transmission group. You can assign up to 255 SDLC, Token-Ring, and frame-relay logical lines in any combination to the same transmission group. Code TGCONF=MULTI or omit TGCONF on the PU definition statement for all link stations in the transmission group.

For a multilink transmission group that contains at least one Token-Ring or frame-relay line, *both* ends of the link must be NCP V6R2 or a later release; otherwise, link activation will fail.

ESCON Links: TGN is not valid for ESCON physical link stations.

TIMEOUT
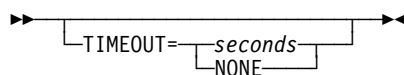
```
►►──────┬────────────────────────────┬──►◄
        └─TIMEOUT=──┬─seconds─┬───────┘
                    └─NONE────┘
```

Use: ESCON logical link stations
Default: See text
Minimum: 10.0
Maximum: 840.0

Specifies how long NCP is to wait for a response to an attention signal it sent to the host before initiating channel discontact (TIMEOUT=*seconds*), or that NCP is to wait an unlimited time for a response from the host (TIMEOUT=NONE).

Specify *seconds* either as an integral number of seconds or to the nearest 10th of a second. If you code DOS, VSE, OS, or MVS for TYPSYS on the BUILD definition statement, the default for TIMEOUT is 180.0 seconds. If you code TYPSYS=VM, the default for TIMEOUT is 420.0 seconds.

Attention: Specifying TIMEOUT=NONE results in non-responding terminals, long-term slowdown of the communication controller, and blocked virtual routes through the network if contact with the host is lost. You should not specify TIMEOUT=NONE if you have a complex network.

TIMERS
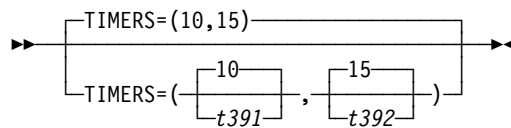
```
          ┌─TIMERS=(10,15)──────────────┐
►►────────┤                             ├────►◄
          │            ┌─10─┐     ┌─15─┐ │
          └─TIMERS=(───┤    ├──,──┤    ├─)┘
                       └─t391─┘   └─t392─┘
```

Use: Frame relay (link-station subport only)
                    ISDN (physical PU only)
                    NTRI frame handler logical line (link-station subport only)

Specifies the length of the T391[13] and T392[13] time intervals used by the frame-relay LMI protocol. TIMERS is valid only when this PU definition statement specifies an LMI standard (not LMI=NO).

t391

    Default: 10
    Minimum: 5
    Maximum: 30

    Specifies the T391 interval, that is, the interval, in seconds, between trans-missions of LMI status enquiry messages.

t392

    Default: 15
    Minimum: 5
    Maximum: 30

    Specifies the T392 interval, that is, the interval, in seconds, in which an LMI status enquiry message must be received before the time interval is considered an error interval.

The value of *t391* must be less than the T392 interval for the adjacent frame-relay node, and the value of *t392* must be greater than the T391 interval for the adjacent frame-relay node. Otherwise, the frame-relay or ISDN link-station subport (LMI station) or NTRI frame handler link station subport may fail. An LMI threshold error (see ERRORT on the PU definition statement on page 2-380) is no longer a link

---

[13] T391 and T392 are standard frame-relay values described by CCITT standard Q.933 Annex A and ANSI standard T1.617 Annex D.

error but a station error. It is recommended that you use the same values for TIMERS on both ends of the line.

Notes:

1. NCP can lose contact with an adjacent communication controller over frame-relay lines when the controller is actively loading or dumping over the link. To prevent a time-out error, code the TIMERS and ERRORT keywords on the first PU definition statement associated with the frame-relay physical line definition (if the LMI keyword is also coded, and not LMI=NO). Specify a minimum retry time of 3 minutes before an LMI threshold error occurs.

2. For NCP V7R2 and earlier releases, NCP will issue a +RSP(IPLFINAL) to VTAM to indicate that the load is complete. Because the remote NCP cannot respond to LMI messages during the load process, frame-relay physical lines attached to the remote NCP may become inoperative (INOP 02) after the load is complete. In this event, you need to reactivate the physical lines. For NCP V7R3 and later releases, the physical line no longer INOPs. The logical FRTE line may become inoperative (INOP 02) and the LMI link-station may also become inoperative (INOP 01). In this event, you need to reactivate the logical lines.

   To avoid the condition where an INOP occurs during a remote load, the LMI error threshold time interval for the link station subport should be at least 3 times the T1 timer interval for the logical subarea line. The LMI error threshold time interval is $t391 \times n392$, where $t391$ is a time interval defined by TIMERS on the PU definition statement for the link station subport.

   If you do not code the time intervals as just described, the LMI PU and associated logical lines over which NCP is being loaded may become inoperative (INOP 01 and INOP 02) respectively *before* NCP issues the +RSP(IPLFINAL) to VTAM, that is, before the load is complete. In this event, the load will fail.

   See the *NCP, SSP, and EP Resource Definition Guide* for more information about how to code the TIMERS keyword when the line will be used for remote loading of an NCP.

3. When a physical frame-relay line or ISDN line using the LMI protocol has a large number of DLCIs, the T391 and T392 timer values must be adjusted for delays caused by low line speeds. Since all frames, including LMI frames, are queued by NCP, a large volume of traffic can create a delay that exceeds the default value for the T391 timer. To avoid a time-out, increase the TIMERS value at each end of the line.

TRANSFR

```
►►──┬──────────────────┬──►◄
    └─TRANSFR=count─┘
```
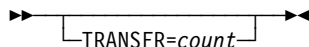
Use:	ESCON physical link stations
Default:	Value of TRANSFR on the BUILD definition statement
Minimum:	1
Maximum:	254

Specifies the number of NCP buffers corresponding to the maximum amount of data that NCP can receive from the station (the data transfer limit) during a single data transfer operation.

T1TIMER
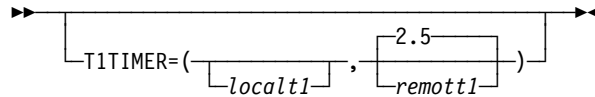
Use: Token-Ring logical subarea resources
Frame-relay logical resources
ISDN logical resources

Specifies, in seconds, the reply timer (T1) values for stations connected to ISDNs, Token-Ring networks, or frame-relay networks.

For Token-Ring and frame-relay resources, you can also specify these values using LOCALTO and REMOTTO. See LOCALTO on the PU definition statement on page 2-389 and REMOTTO on the PU definition statement on page 2-413 for specific information.

localt1

Default: See text
Minimum: 0.6
Maximum: 20.0

Specifies the reply timer value for stations connected to ISDNs, Token-Ring networks, or frame-relay networks.

Token Ring: *localt1* specifies the Token-Ring reply timer (T1) value for stations connected to a local Token Ring. *localt1* is valid only for logical subarea lines. (Code *localt1* on T1TIMER on the BUILD definition statement to specify a T1 value for logical peripheral lines.)

The value of *localt1* must not be greater than the value of *remott1*. The default value for *localt1* is either the value of *remott1* on this T1TIMER on the LINE definition statement or the value of *remott1* on T1TIMER on the BUILD definition statement, whichever is less.

Frame Relay: *localt1* specifies the frame-relay reply timer value for stations connected to a frame-relay network. If you do not code *localt1* on either the LINE or GROUP definition statement, the default is 6.0. The value of T1TIMER on the BUILD definition statement is not used for frame relay.

Important: Code a value for *localt1* that reflects the maximum amount of time that it takes to receive a solicited response from the adjacent link station. Factors that can affect this time include line speed, propagation delay, maximum frame size, delay caused by intermediate frame handlers, and the number of DLCIs supported by the associated physical line that could create queuing delays for the FRTEs. If the value you code for *localt1* is too small, the FRTEs can lose synchronization and fail the connection. This is indicated by an INOP 01 and a link event or an alert that indicates a FRMR condition. If this occurs, code a larger *localt1* value for this FRTE connection.

If a substitute route for a PVC is defined in the frame-relay network, you may need to code the values of *localt1* and RETRIES such that the combination of these two values is greater than the LMI error threshold

of any node in the PVC. This gives the network time to switch to the substitute PVC.

An indication of INOP 02 can occur for a frame-relay logical line if it is used to perform a remote load. To avoid the condition where the INOP 02 occurs prior to the +RSP(IPFINAL), the LMI error threshold time interval for the link station subport should be at least 3 times the T1 timer interval for the logical subarea line. The LMI error threshold time interval is $t391 \times n392$, where $t391$ is a time interval defined by TIMERS on the PU definition statement for the link station subport, and $n392$ is an error count threshold defined by ERRORT on the same PU definition statement.

NTRI Frame Handler: *localt1* is *not* valid for NTRI frame handler logical lines. (Code *localt1* on T1TIMER on the BUILD definition statement to specify a *localt1* value for NTRI frame handler logical lines.)

ISDN: *localt1* specifies the ISDN reply timer value for stations connected to an ISDN. If you do not code *localt1* on either the LINE or GROUP definition statement, the default is 6.0. The value of T1TIMER on the BUILD definition statement is not used for ISDN.

Important: Code a value for *localt1* that reflects the maximum amount of time that it takes to receive a solicited response from the adjacent link station. Factors that can affect this time include line speed, propagation delay, maximum frame size, and the number of DLCIs supported by the associated physical line that could create queuing delays for the adjacent link station. If the value you code for *localt1* is too small, the adjacent link stations can lose synchronization and fail the connection. This is indicated by an INOP 01 and a link event or an alert that indicates a FRMR condition. If this occurs, code a larger *localt1* value for this adjacent link station connection.

remott1

Default: 2.5
Minimum: 0.6
Maximum: 20.0

Specifies, in seconds, the reply timer (T1) value for stations connected to a remote Token Ring. You can specify this value as an integer or with one decimal place.

Token Ring: *remott1* is used for Token-Ring logical subarea resources only.

Frame Relay: *remott1* is not used for frame-relay logical resources.

ISDN: *remott1* is not used for logical resources.

T2TIMER

```
►►──┬─────────────────────────────────────────────────┬──►◄
    └─T2TIMER=(──┬────────┬──,──┬────────┬──,──┬────┬──)─┘
                 └─localt2─┘     └─remott2─┘     └─n3─┘
```
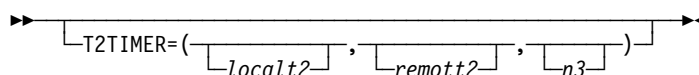
Use: Token-Ring logical subarea, frame-relay and ISDN logical lines

Specifies the acknowledgment timer (T2) values and the maximum number of I-frames that NCP can receive without sending an acknowledgment. This keyword

is valid only if you code ECLTYPE=(LOGICAL,SUBAREA), FRELAY=LOGICAL, ISDN=(LOGICAL,SUBAREA), or ISDN=(LOGICAL,PERIPHERAL) on the GROUP definition statement.

If possible, consider the value of the reply timer (T1) used by the adjacent station when determining the value for the receiver acknowledgment timer. The acknowledgment timer should be about one-tenth as long as the reply timer for the adjacent station to ensure that the adjacent station will receive the delayed acknowledgment before its reply timer expires.

T2TIMER can be coded on the PU definition statement for frame-relay and ISDN logical lines and Token-Ring logical subarea lines.

localt2

>
> Default: See text
> Minimum: 0.0
> Maximum: 2.0

> **Token Ring:** Specifies the timer value used for connections over a local Token Ring. Valid values are in intervals of a tenth of a second. The default is the value coded for *localt2* on the BUILD definition statement. A value of 0 indicates that the timer is not to be used for this line.

> **Frame-Relay Logical Lines:** Specifies the timer value for connections through a frame-relay network. For logical subarea lines, FRELAY=(LOGICAL,SUBAREA) on the GROUP definition statement, the default is 1.0 if the line number of the corresponding physical line is in the range 0–895; the default is 0.2 if the line number of the corresponding physical line is in the range 1024–1039. For frame-relay logical peripheral lines, the default is 0.0.

> > The *localt2* value on the BUILD definition statement is not the default used for frame-relay lines.

> > A value of 0.0 indicates that the timer is not to be used for this line.

> **3746 Model 900 Frame-Relay Logical Lines:** For logical subarea lines, if the speed of the corresponding physical line is greater than or equal to 256KB, the default value is 0.2. If the corresponding physical line speed is less than (<) 256KB, the default value is 1.0. For logical peripheral lines the default value is 0.0.

> **ISDN:** The default value is 1.0 for ISDN logical subarea lines and 0.0 for ISDN logical peripheral lines.

remott2

>
> Default: See text
> Minimum: 0.0
> Maximum: 2.0

> **Token Ring:** Specifies the timer value used for connections over a remote Token Ring. Valid values are in intervals of a 10th of a second. The default value is the value coded for *remott2* on the BUILD definition statement. A value of 0 indicates that the timer is not to be used for this line.

> **Frame Relay:** *remott2* is not valid for frame relay.
>
> **ISDN:** *remott2* is not valid for ISDN.

n3

> Default: See text
> Minimum: 1
> Maximum: 127
>
> **Token Ring:** Specifies the maximum number of I-frames that can be received
> without sending an acknowledgment. The default is the value coded for
> *n3* on the T2TIMER keyword on the BUILD definition statement.
>
> **Frame-Relay and ISDN Logical Lines:** Specifies the number of I-frames that
> can be received without sending an acknowledgment. If *n3* is not
> coded on either the LINE or the GROUP definition statement, the
> default is 2, not the value coded on the BUILD definition statement.

UNIQUE

```
             ┌─UNIQUE=YES─┐
►►───────────┼────────────┼───────►◄
             └─UNIQUE=─┬─YES─┬──────
                       └─NO──┘
```

Use: Token-Rink logical subarea links.
Default: YES

Specifies whether the 12-character medium access control (MAC) address, which is
defined for the ADDR keyword and corresponds to the physical token-ring con-
nection in the adjacent node, is a unique locally administered address.

Code UNIQUE=YES if the 12-character MAC address is a unique locally adminis-
tered address. This will prompt a test frame with a zero DSAP value to be sent
during route discovery processing.

Code UNIQUE=NO if the 12-character MAC address is a duplicate locally adminis-
tered address, that is, duplicate TIC addressing for logical subarea connections.
This will prompt a test frame with the DSAP value from the ADDR keyword to be
sent during route discovery processing.

For additional information on ADDR, refer to the ADDR keyword on the PU defi-
nition statement on page ADDR on the PU definition statement on page 2-366.

See the *NCP, SSP, EP Resource Definition Guide* for more information about
defining Token-ring logical subarea links.

XID

```
        ┌────────────┐
►►──────┴─XID=─┬─YES─┬──────►◄
               └─NO──┘
```
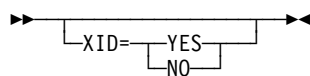
Default: See text

Specifies whether this physical unit can receive and respond to an XID while in
normal disconnected mode (NDM). Code XID=YES for all physical units that

process XID in NDM. You must use YES for type 2.1 physical unit nodes. If you code XID=NO, an XID is not sent to the physical unit.

XID=NO is not valid for switched lines (DIAL=YES on GROUP).

For casual connection, specify XID=YES when MODE=SECONDARY on the GROUP definition statement.

APPN: You need to define type 2.1 physical units to support APPN functions for NCP. Code PUTYPE=2 and XID=YES.

Table 2-107 shows the effect on XID when you code certain keywords on the GROUP definition statement.

Table 2-107. NDF Values for XID on the PU Definition Statement

When You Code This on GROUP	The Effect on XID Is
DIAL=YES	The default is XID=YES. The only valid value is XID=YES.
DIAL=NO	The default is XID=NO.
LNCTL=CA	The default is XID=YES. The only valid value is XID=YES.
ECLTYPE=PHYSICAL	The default is XID=NO. The only valid value is XID=NO.
ECLTYPE=(LOGICAL,SUBAREA)	The default is XID=NO.
NCST=IP	The default is XID=NO.
ETHERNET=PHYSICAL	The default is XID=NO.
FRELAY=LOGICAL	The default is XID=NO.
FRELAY=PHYSICAL and DIAL=NO	The default is XID=NO. The only valid value is XID=NO.

XMONLNK
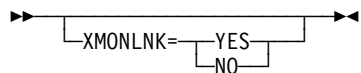
Default: See text

Specifies whether the SMMF is to initially activate this link station. If you code XMONLNK=NO, VTAM must activate the link station. For more information, see the discussion of SMMF in the *NCP and EP Reference*.

XMONLNK=YES is valid only for peripheral link stations, so it is not valid if you code PUTYPE=4 or PUTYPE=5. When LNCTL=SDLC, XMONLNK=YES is valid only when VIRTUAL=NO and DIAL=NO, with one exception. The exception is switched frame relay, where XMONLNK=YES is allowed when DIAL=YES and FRELAY=(PHYSICAL,SUBAREA) or FRELAY=(PHYSICAL,ANY).

Note: Be sure to spell XMONLNK without the letter *I*.

Lines Attached to a 3745 Low-Speed Scanner: The scanner scans only LICs with active lines, to increase efficiency. Coding XMONLNK=YES can cause a line to become active, thus increasing the number of lines that the

scanner must scan. Care must be taken not to overload the scanner. For information about scanner capacities, see *3745 Communication Controller, All Models: Connection and Integration Guide*, SA33-0129.

ESCON Links: XMONLNK=YES is not valid for ESCON logical link stations. Code XMONLNK=YES for an ESCON physical link station if MONLINK=YES or MONLINK=CONTINUOUS on the LINE definition statement for a logical link associated with the physical link.

To load or dump NCP over an ESCON link, do the following:

- Code MONLINK=YES or MONLINK=CONTINUOUS on the LINE definition statement for the logical line.

- Code MONLINK=YES on the PU definition statement for the logical link station you want to use.

- Code XMONLNK=YES on the PU definition statement for the physical link station.

Spare or Redefinable Lines: XMONLNK=NO is the default. XMONLNK=YES is not valid.

Table 2-108 shows the effect on XMONLNK when you code certain keywords on the BUILD and GROUP definition statements.

Table 2-108. NDF Values for XMONLNK on the PU Definition Statement

When You Code This on GROUP	The Effect on XMONLNK Is
ECLTYPE=PHYSICAL	See Token Ring note following.
FRELAY=PHYSICAL	See Frame Relay note following.
ISDN=PHYSICAL	See ISDN note following.
ESCON physical link station	XMONLNK=YES is the default.
ETHERNET	The default is XMONLNK=NO. The only valid value is XMONLNK=NO.

Token Ring: The default is XMONLNK=YES when TYPGEN=NCP-R on the BUILD definition statement, and ECLTYPE=(PHYSICAL,SUBAREA) or ECLTYPE=(PHYSICAL,ANY). Otherwise, the default is XMONLNK=NO.

Frame Relay: For the first PU definition statement that defines the link station subport on a physical line, the default is XMONLNK=YES if TYPGEN=NCP-R and FRELAY=(PHYSICAL,SUBAREA) or FRELAY=(PHYSICAL,ANY); otherwise, the default is XMONLNK=NO. For PU definition statements that define a FHSP, the default is XMONLNK=NO. For PU definition statements that define a terminating equipment subport (TESP), or any PU definition statement in a frame-relay logical group (FRELAY=LOGICAL), XMONLNK is not valid.

ISDN: For the ISDN physical line PU or an ISDN B-channel PU, the default is XMONLNK=YES if TYPGEN=NCP-R; otherwise, the default is XMONLNK=NO. For any PU definition statement in an ISDN logical group, XMONLNK is not valid.

Code XMONLNK=YES for a physical link station if MONLINK=YES or MONLINK=CONTINUOUS on the LINE definition statement for a logical subarea line associated with the physical line.

Code XMONLNK=YES for a NTRI physical link station if MONLINK=YES is coded for a NTRI frame handler logical link station associated with the physical line.

PUDRPOOL Definition Statement

Usage: Dynamic reconfiguration.

Position: Must precede the first GROUP definition statement.

Keywords: Table 2-109 shows the keywords described under this definition statement.

Table 2-109. Keywords on the PUDRPOOL Definition Statement

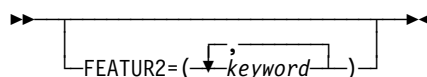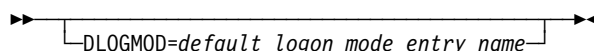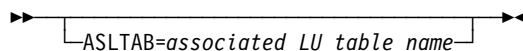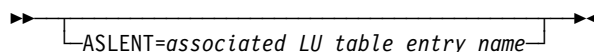| Keyword	Page	Required	Sift	Default	Tuning*
FRSEDRPU	2-429				
NUMBER	2-430			0	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.
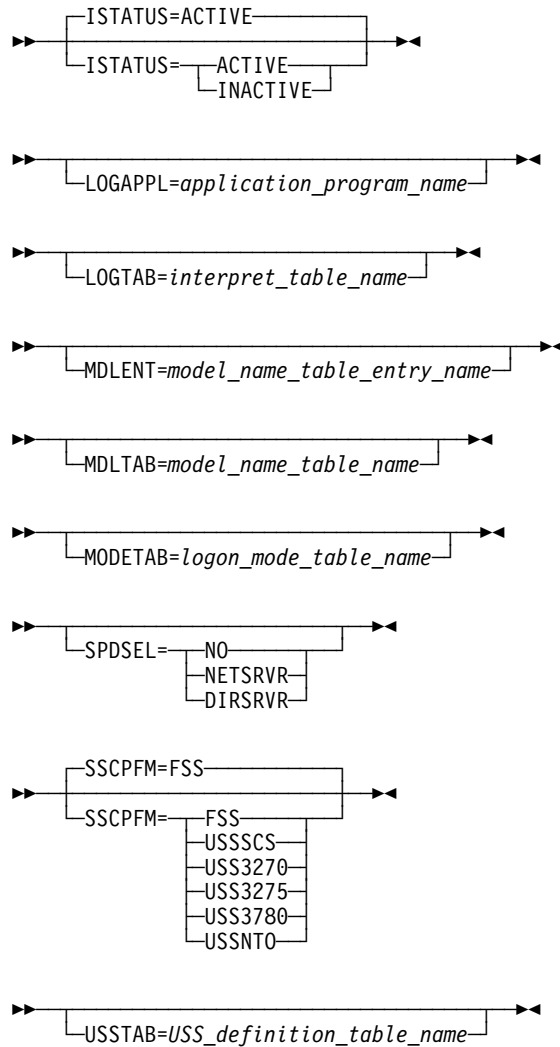
Keyword Descriptions

symbol

►►—*symbol*—►◄

Use: Required when you code the PUDRPOOL definition statement
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

FRSEDRPU

►►──┬───────────────────┬──►◄
    └─FRSEDRPU=─┬─*count*──┬─┘
               └─*frdrcnt*─┘

Use: Frame-relay frame handler subports
            NTRI frame handler logical subports
Default: Value of the NUMBER keyword on the PUDRPOOL definition statement

count

   Minimum: **0**
   Maximum: **32766**

Specifies how many 3745 frame handler subports added to NCP through dynamic reconfiguration are to be eligible for NPM performance data collection.

frdrcnt
Specifies how many 3745 frame-relay physical units added to NCP through dynamic reconfiguration.

You should specify *frdrcnt*, but its default is the value of the NUMBER keyword on the PUDRPOOL definition statement. Its maximum value is the same as the NUMBER keyword or 16 381. The *frdrcnt* value must also be less than, or equal to, the value of NUMBER.

In addition, because *frdrcnt* is a general frame-relay DR parameter, the value of the *count* operand (which is restricted to NPA) should never exceed the *frdrcnt* value. This means, if *frdrcnt* is specified and *count* is not, *count* does not take its normal PUDRPOOL NUMBER default, but uses the value of *frdrcnt* instead.

FRSEDRPU is valid only if you specify NPA=(YES,DR) or NPA=(YES,DRTP) on the BUILD definition statement.

3746 Model 900 Frame Relay: 3746 Model 900 frame-relay resources do not use the storage reserved by FRSEDRPU. Therefore, do not include these resources when calculating the value of FRSEDRPU.

For V7R1 or later, the value of FRSEDRPU can be greater than the value of NUMBER if the DYNPOOL keyword is coded. If the DYNPOOL keyword is not coded, the value of FRSEDRPU must be less than or equal to the value of NUMBER. For V6R3 or earlier, the value of FRSEDRPU must always be less than or equal to the value of NUMBER.

Note: A large FRSEDRPU value can waste NCP storage. If some of the physical units in the pool will *not* be used for frame relay, code FRSEDRPU to specify how many *will* be used for frame relay. This requires less storage than allowing FRSEDRPU to default to the value of NUMBER.

NUMBER

```
          ┌─NUMBER=0────────┐
►►────────┼─────────────────┼────►◄
          └─NUMBER=count────┘
```
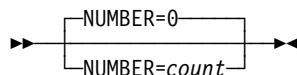
Default: 0
Minimum: 0
Maximum: 32766

Specifies the number of physical units to be included in the physical unit pool. To conserve controller storage, base *count* on the anticipated dynamic resource requirements.

Notes:

1. For performance reasons, code a value greater than 0 if you are using dynamic reconfiguration to free and add physical units. To avoid freeing a physical unit and immediately adding it back with an RNAA, do not allow the physical unit pool to become empty.

2. You can define part of the NCP buffer pool to be used for dynamically allocated

physical units when explicitly defined physical units have been exhausted. For
more information, see DYNPOOL on the BUILD definition statement on
page 2-19. Since overuse of dynamic resources can degrade NCP perform-
ance, define all the resources required for normal processing in your generation
definition and use this function only to provide control block resources for unex-
pected demand. See the *NCP, SSP, and EP Resource Definition Guide* for
more information about dynamically created control blocks.

REMOVCTL Definition Statement

Usage: Start-stop devices only. There is one REMOVCTL definition statement allowed for each block handler. Use the REMOVCTL definition statement to specify whether NCP removes the ending control characters from incoming terminal data blocks. This definition statement removes characters from control blocks in which an ending sequence was detected. A block handler routine must be specified to execute at point 2.

Position: Can appear in any sequence between the STARTBH and ENDBH definition statements.

Keywords: This definition statement has no keywords.
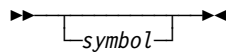
symbol

```
►►─┬─────────┬─►◄
   └─symbol──┘
```

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

SDLCST Definition Statement

Usage: Subarea links connecting two communication controllers in a point-to-point configuration. There is one definition statement for NCP when it is in a primary state, and one for NCP when it is in a secondary state. If you code a symbol on the SDLCST definition statement, you must also code the SDLCST keyword on the GROUP or LINE definition statement.

Token Ring: You should code both a primary and a secondary SDLCST definition statement for Token-Ring logical subarea lines. At least one is required.

Only the GROUP, MODE, and MAXOUT keywords apply to all Token-Ring resources. Additionally, the RETRIES keyword applies to 3746 Model 900 Token-Ring resources and overrides the *m,t,n* values coded on the RETRIES keyword on the LINE statement. The *ml* value coded on the RETRIES keyword on the LINE statement is not overridden and will be used for retries on a local ring.

Frame Relay: You should code both a primary and a secondary SDLCST definition statement for frame-relay logical subarea lines. At least one is required. The only keywords listed below that apply to frame relay are GROUP, MODE, and RETRIES.

ISDN: You should code both a primary and a secondary SDLCST definition statement for ISDN logical subarea lines. At least one is required. The only keywords listed below that apply to ISDN are GROUP, MODE, and RETRIES.

Position: Must precede first GROUP definition statement.

Keywords: Table 2-110 shows the keywords described under this definition statement.

Table 2-110. Keywords on the SDLCST Definition Statement

Keyword	Page.	Required	Sift	Default	Tuning*
GROUP	2-434	Yes			
MAXOUT	2-434			7	Yes*
MODE	2-435			PRIMARY	
PASSLIM	2-435			254	Yes*
PAUSE	2-435				Yes*
RETRIES	2-436			(15,0,1)	
SERVLIM	2-437			4	Yes*
TADDR	2-438				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the SDLCST definition statement
Default: None

Provides a name for the line control selection table entry. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

GROUP

►►──GROUP=*name*──►◄

Use: Required when you code the SDLCST definition statement
Default: None

Specifies the name of the GROUP definition statement that contains the link parameters associated with this SDLC selection table entry.

The keywords on the GROUP definition statement associated with the table entry are LNCTL, DIAL, ACTIVTO, and REPLYTO. The GROUP definition statement must specify LNCTL=SDLC and DIAL=NO. The GROUP and SDLCST definition statements must both specify MODE=PRIMARY or MODE=SECONDARY.

name is the the name of either a GROUP definition statement that is followed by LINE and other lower-level definition statements, or a stand-alone GROUP definition statement.

MAXOUT

```
              ┌─MAXOUT=7──────┐
►►────────────┼───────────────┼──►◄
              └─MAXOUT=frames─┘
```

Use: Secondary mode
Default: 7
Minimum: See text
Maximum: See text

Specifies the maximum number of information frames (I-frames) that can be sent to an adjacent Type 4 link station when an acknowledgement for the frames sent is expected but is not received.

The valid range for *n* is 1 to 127. For lines operating as modulo-8 lines, specify a value from 1 to 7. For lines operating as modulo-128 lines, specify a value from 8 to 127.

For a complete description of the MAXOUT keyword as it applies to link stations in adjacent Type 4 nodes, see MAXOUT on the PU definition statement on page 2-395.

MODE

```
                  ┌─MODE=PRIMARY────────┐
►►────┼─────────────────────┼────►◄
                  └─MODE=─┬─PRImary───┬─┘
                               └─SECondary─┘
```
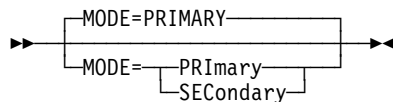
Default: PRIMARY

Specifies whether this selection table entry is to be used when the controller is functioning as a primary or a secondary device. MODE determines the mode of the line group specified by the GROUP keyword.

Code the same value for MODE here and on the associated GROUP definition statement.

PASSLIM

```
         ┌─PASSLIM=254──┐
►►────┼──────────────┼────►◄
         └─PASSLIM=pius─┘
```
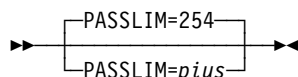
Default: 254
Minimum: 1
Maximum: 254

Limits the maximum number of consecutive path information units (PIUs) that can be sent to an adjacent link station when that link station is due to receive PIUs.

For a complete description of the PASSLIM keyword as it applies to link stations in adjacent Type 4 nodes, see PASSLIM on the PU definition statement on page 2-404.

PAUSE

```
►►──┬──────────────────────────────────────┬──►◄
     └─PAUSE=(─┬───────────┬─┬──────────────┬─)─┘
                   └─modulo-8─┘ └─,modulo-128─┘
```
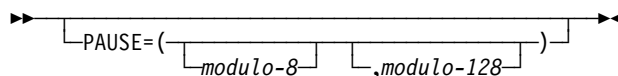
Default: Value of PAUSE on the LINE definition statement
Minimum: 0
Maximum: 25.5

Specifies, in seconds, the average duration of the polling cycle for SDLC primary or secondary subarea links. Use the PAUSE keyword to specify a different value for primary and secondary subarea links.

The polling cycle begins when NCP examines the first entry in the SOT and ends when polling begins again at the same entry. If the SDLC link is in a secondary state, the PAUSE keyword specifies the time NCP can defer answering a poll from the adjacent primary NCP.

modulo-8
> Specifies the pause for this line when it is operating as a modulo-8 line.

modulo-128
> Specifies the pause for this line if the line is operating as a modulo-128 line.

For large subarea configurations (for example, over 500 lines), the PAUSE value for the secondary state must be at least 0.5 second.

RETRIES
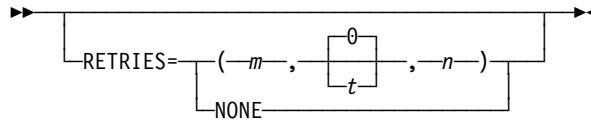
Specifies how many attempts NCP is to make to recover from errors occurring during transmission over the link associated with this SDLCST definition statement. Each attempt at recovery, or *retry*, is a retransmission of data or control commands. NCP performs retries until the transmission is successful or the number of retries reaches the limit specified by this keyword.

Retries are performed in sets, which are called *retry sequences*. NCP can perform one or more retry sequences. Successive retry sequences are separated by pauses. You can specify the number of retries in a sequence (m), the number of sequences to be performed (n), and the length of the pause between sequences (t). The maximum number of retries NCP attempts equals $(m + 1) \times (n + 1) - 1$.

If you code RETRIES=NONE or RETRIES=0, NCP performs no retries at all.

Attention: If the values you specify for RETRIES on this definition statement and REPLYTO on the GROUP definition statement result in a retry time of less than 3 minutes, contact may be lost with an adjacent link-attached controller.

m

> Default: See text
> Minimum: 0
> Maximum: 128

> Specifies the maximum number of retries in each retry sequence.

> When MODE=PRIMARY, the default is 7. When MODE=SECONDARY, the default is 15.

t

> Default: 0
> Minimum: 0
> Maximum: 255

> Specifies the pause interval, in seconds, between successive retry sequences. You can specify a value for t that is greater than zero only when the value of m is greater than zero.

n

Default:	See text
Minimum:	1
Maximum:	127

Specifies the maximum number of retry sequences.

If *m* is less than or equal to 64, and *t* is not equal to 0, the default is 1; otherwise, the default is 0.

For more information, see RETRIES on the LINE definition statement on page 2-269.

Frame Relay: If a substitute route for a PVC is defined in the frame-relay network, you may need to code the values of LOCALTO and RETRIES such that the combination of these two values is greater than the LMI error threshold of any node in the PVC. This gives the network time to switch to the substitute PVC.

SERVLIM
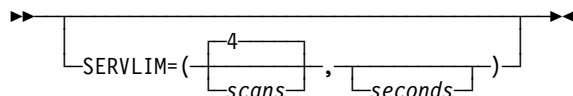
Determines how often NCP interrupts its regular scans of the service order table (SOT).

scans

Use:	Primary SDLC subarea links
Default:	4

Specifies how many regular scans of the SOT NCP is to make for normal servicing of the physical unit on the link before it makes a special scan of the table. The regular scan of the table accommodates normal transmission of PIUs between the access method or host application programs and the physical unit on the link.

For more information, see SERVLIM on the LINE definition statement on page 2-274.

seconds

Use:	Primary SDLC subarea links
Default:	See text
Minimum:	1.0
Maximum:	600.0

SDLC Links: Specifies the minimum time between special scans of the service order table.

The default is none, which means that the frequency of special scans will be determined by *entries*. If you specify both *seconds* and *entries*, *seconds* determines the frequency of special scans.

TADDR

```
►►──┬─────────────────┬──►◄
     └─TADDR=chars─┘
```
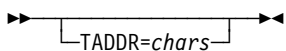
Use: Secondary SDLC subarea links
Default: See text

Specifies the 2-digit hexadecimal representation of a single EBCDIC SDLC device address, that is, the address of the secondary controller in which NCP resides. You can assign any bit configuration except X'00' or X'FF' as the address. TADDR is valid only if MODE=SECONDARY.

The default for TADDR is the rightmost 2 digits of the hexadecimal representation of the subarea address coded on the SUBAREA keyword on the BUILD definition statement. There are two exceptions to this rule. If the rightmost 2 digits are X'00', the default is X'01'. If the rightmost 2 digits are X'FF', the default is X'FE'. The examples in Table 2-111 illustrate the calculation of the default.

Table 2-111. Examples of TADDR Defaults on SDLCST Definition Statement

SUBAREA on BUILD	Hexadecimal Equivalent	Rightmost 2 Digits	TADDR Default
1	0001	01	01
12346	303A	3A	3A
4608	1200	00	01
4607	11FF	FF	FE

SERVICE Definition Statement

Usage: Nonswitched telecommunication links only; one for each telecommunication link except for subarea links connecting two NCPs. For a line operating in network control mode, NDF automatically generates the SOT if you omit the SERVICE definition statement or code SERVICE and omit the ORDER keyword.

Casual Connection: If you omit the SERVICE definition statement and MODE=SECONDARY on the GROUP definition statement, the SOT will contain one entry.

ESCON Links: The SERVICE definition statement is not valid for ESCON links.

Position: Must follow the LINE definition statement.

Keywords: Table 2-112 shows the keywords described under this definition statement.

Table 2-112. Keywords on the SERVICE Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
MAXLIST	2-440				
ORDER	2-440				

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions
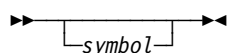
symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates. If you specify more than one SERVICE definition statement, code *symbol* only on the first one.

symbol is required for BSC and start-stop lines.

MAXLIST

```
►►──┬──────────────────┬──►◄
    └─MAXLIST=entries──┘
```
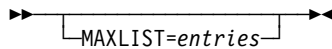
Default: See text
Minimum: 1
Maximum: 254

Specifies the maximum number of entries in the SOT. If the value of MAXLIST is greater than the value of ORDER on this SERVICE definition statement, you can add more entries (up to the MAXLIST limit) during NCP execution through dynamic reconfiguration. If the value of MAXLIST is less than the value of ORDER, NCP uses the value of ORDER for MAXLIST.

If you omit MAXLIST, NCP uses the number of entries specified by ORDER, or the number of resources if you do not code ORDER. No further entries can be added or moved while NCP is running. (However, you can change entries with the dynamic reconfiguration function.)

MAXLIST is not valid if you specify MODE on the GROUP definition statement and do not code SDLCST on GROUP for multipoint subarea links. For casual connection, code MAXLIST=1 when MODE=SECONDARY on the GROUP definition statement.

Ethernet-Type LANs: If ETHERNET=PHYSICAL is specified on the GROUP definition statement, only MAXLIST=1 is valid. The default is 1.

3745 Frame Relay: If FRELAY=PHYSICAL is specified on the GROUP definition statement, only MAXLIST=1 is valid. The default is 1.

ORDER

```
►►──┬─────────────────────────┬──►◄
    │         ┌─,─────┐        │
    └─ORDER=(─▼─device─┴──)─┘
```
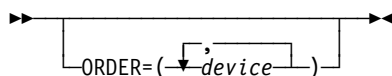
Specifies the order in which the devices on the line are to be serviced. *device* is the name of a device, clustered terminal, component, or link-attached communication controller. These names appear on the TERMINAL, CLUSTER, COMP, or PU definition statements associated with the line for which this SOT is being generated. A device or component name can appear in the list more than once.

You can code a maximum of 255 characters in ORDER, including the beginning and ending parentheses and all commas. This limit applies regardless of how many entries you code for ORDER. If you need to specify more than 255 characters, code additional SERVICE definition statements following the first SERVICE definition statement. Omit MAXLIST in the additional definition statements. Specify the remaining entries on the ORDER keywords. The maximum number of entries is 254.

If you omit ORDER, NDF automatically generates the SOT for a line operating in network control mode. The order of the values is the same order in which the physical units are specified.

The generation process checks the entries of the SOT to determine if the devices represented by the CLUSTER, TERMINAL, COMP, or PU definition statement are

associated with this line. Also, each CLUSTER, TERMINAL, COMP, and PU defi-
nition statement associated with the LINE definition statement is checked to deter-
mine if it is represented in the SOT. For user line control, ORDER and MAXLIST
default as described only if the SERVICE definition statement is coded.

Casual Connection: Specify only one entry for ORDER when
MODE=SECONDARY on the GROUP definition statement.

Automatic Line Generation: ORDER is not valid if you code AUTOCOPY on the
LINE definition statement.

Ethernet-Type LANs: If you specify ORDER, you may define only one entry.

3745 Frame Relay and 3745 Token Ring: ORDER is valid only for frame-relay
physical lines and NTRI frame handler logical lines. You can specify only
the name of the link-station subport (the first PU definition statement under
the LINE definition statement for a frame-relay physical line or a NTRI frame
handler logical line). If you do not code ORDER, the SOT is automatically
generated with the link-station subport as its only entry. ORDER for 3745
Token Ring dictates to NDF how to build the SOT with regards to the phys-
ical stations.

3746 Model 900 Frame Relay: The ORDER keyword is ignored.

3746 Model 900 SDLC Lines: For 3746 Model 900 SDLC lines, ORDER deter-
mines the relative frequency at which the devices are polled, but not the
polling order. For example, if you code ORDER=(A,B,A,C), the 3746 Model
900 polls device A twice as often as B or C, though not necessarily in that
order. As a result, the order in which you specify the devices does not
matter; ORDER=(A,B,A,C) has the same effect as ORDER=(A,A,B,C).

STARTBH Definition Statement

Usage: BSC or start-stop devices only. One for each block handler.

Position: Must appear at the beginning of the block handler.

Keywords: Table 2-113 shows the keyword described under this definition state-ment.

Table 2-113. Keywords on the STARTBH Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
BHEXEC	2-442			PT1	

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►──*symbol*──►◄

Use: Required when you code the STARTBH definition statement
Default: None

Provides a name for the block handler set. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

BHEXEC

```
                 ┌─BHEXEC=PT1───────┐
►►───────┼──────────────────┼───►◄
                 └─BHEXEC=─┬─PT1─┬──┘
                          ├─PT2─┤
                          └─PT3─┘
```

Default: PT1

Specifies the point at which this block handler is to run.

If you specify BHEXEC=PT1, NCP invokes this block handler upon receiving a contact or write request from the host processor, but before it determines whether the line is available to contact the device.

Data Sent to a Device: If you want the data received from the access method processed after NCP has contacted the device, code BHEXEC=PT2.

Attention: If you code BHEXEC=PT2, a logical connection between NCP and the device must exist before the block handler is run. The interval

during which the line is not available for communication with other devices is extended by the run time of the block handler.

You can include date and time block handling routines or any user-provided block handling routines in a block handler that processes outgoing data. Therefore, a STARTBH definition statement that specifies BHEXEC=PT1 or BHEXEC=PT2 can be followed by DATETIME or UBHR definition statements.

If the block handler processes incoming data, as well as outgoing data (possible only if you code BHEXEC=PT2), you can also include the EDIT definition statement. However, the block handling routine invoked by the EDIT definition statement processes only the incoming data.

Block handlers process outgoing data only if no data transfer command error occurred.

Data Received from a Device: If you want communication over the line to continue while the block handler is processing the received data, code BHEXEC=PT3.

If you want NCP to suspend further communication over the line until the block handler has processed the received data, code BHEXEC=PT2.

Attention: If you specify BHEXEC=PT2, the line is not available for further communication with devices during the time occupied by block handler execution.

The block handling routines you can include in a block handler that processes incoming data are date and time, edit, or any user-provided block handling routines. Therefore, a STARTBH definition statement that specifies BHEXEC=PT2 or BHEXEC=PT3 can be followed by a DATETIME, EDIT, or UBHR definition statement.

Except for the date and time routine and any user block handling routines, block handlers process incoming data only if no data-check error occurred.

ACCESS on the UBHR definition statement determines whether a user-written routine processes only error-free blocks or only blocks containing errors.

SWMMT Definition Statement

Usage: Defines SMMF tables for switched subarea lines. You must code the SWMMTID keyword on the LINE definition statement to point to the name you assign to the first SWMMT definition statement in the table. Subsequent SWMMT definition statement table entries do not include a name.

Position: Follows the BUILD and SYSCNTRL definition statements. Must appear before the first GROUP definition statement.

Keywords: Table 2-114 shows the keywords described under this definition statement.

Table 2-114. Keywords on the SWMMT Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
IDNUM	2-444	Yes			
MAXDATA	2-445				
MAXOUT	2-445			7	
NETID	2-445				

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

```
►►─────────────────►◄
     └─symbol─┘
```
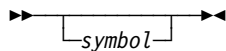
Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

IDNUM

```
►►──IDNUM=line_id──►◄
```
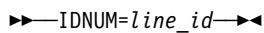
Use: Required when you code the SWMMT definition statement
Default: None
Minimum: 0
Maximum: FFFFF

Specifies a unique identifier for each network for each line for system user verification. Specify the identifier as a hexadecimal number not enclosed in quotation

marks or apostrophes. The identifier must match the identifier specified for the VTAM switched major node physical unit for the calling station.

Use this identifier and the network identifier specified by NETID on this SWMMT definition statement to identify the caller.

MAXDATA

```
►►─┬──────────────┬─►◄
    └─MAXDATA=size─┘
```
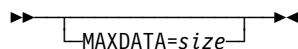
Use: Switched subarea lines within a mixed media multilink transmission
              group
Default: None
Minimum: See text
Maximum: 65535

Specifies the size of the largest path information unit (PIU) or PIU segment that a switched subarea line in a mixed-media multilink transmission group (MMMLTG) sends or receives in one data transfer. See *NCP, SSP, and EP Resource Definition Guide* for information about coordinating PIU sizes within MMMLTGs.

The minimum value of MAXDATA is the value of BFRS on the BUILD definition statement plus 26.

MAXOUT

```
       ┌─MAXOUT=7─────┐
►►─────┼──────────────┼─►◄
       └─MAXOUT=count─┘
```
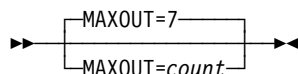
Default: 7
Minimum: 1
Maximum: 127

Specifies the maximum number of PIUs or PIU segments that can be sent to a physical unit before a response is required.

NETID

```
►►─┬────────────┬─►◄
    └─NETID=name─┘
```
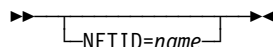
Default: Name of the native network

Specifies the name of the network designated caller. This name must match the name specified by NETID on the NETWORK definition statement that defines the caller's host.

SYSCNTRL Definition Statement

| ***Usage:*** Required. Not applicable to EP standalone. SYSCNTRL specifies the dynamic control facilities for your NCP.

Position: Must follow the BUILD definition statement.

Keywords: Table 2-115 shows the keyword described under this definition statement.

Table 2-115. Keywords on the SYSCNTRL Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
OPTIONS	2-446	Yes			

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

►►─┬──────────┬─►◄
    └─*symbol*─┘

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

OPTIONS

►►──OPTIONS=(─┬─┬─────────┬─┬─)─►◄
            └─◄─*facility*─┘

Use: Required
Default: None

Specifies which dynamic control facilities NCP includes. *facility* can be any value listed in Table 2-116 on page 2-447. Refer to the appropriate access method installation guide or to the *NTO Planning, Migration, and Resource Definition* to determine whether a particular facility applies, is required, or is optional for NCP.

Table   2-116. Dynamic Control Facilities Used by VTAM and TCAM

Value	Facility
BACKUP	Switched network backup
BHSASSC[1]	Modify block handler set
DLRID	Display line resource identifier
DVSINIT	Copy and replace device session initiation information
ENDCALL	Physical disconnect (switched lines only)
LNSTAT	Display line status
MODE	Copy and set destination mode
NAKLIM	Change line negative-polling response limit
RCNTRL	Request control mode reset
RCOND	Reset conditional
RDEVQ	Reset device queues
RECMD	Reset at end of command
RIMM	Reset immediate
SESINIT	Change line session initiation information
SESSION	Change session limit
SSPAUSE	Change service-seeking pause
STORDSP[2]	Display contents of controller storage
XMTLMT	Change device transmission limit

[1] If BHSASSC is coded for a BSC 3270 in session with VTAM, VTAM will disable the block
handler.

[2] STORDSP applies only to non-BSC and non-start-stop NCP definitions. All other
keywords apply only to BSC and start-stop definitions.

TERMINAL Definition Statement

Usage: For BSC or start-stop devices. One for each device including those attached to cluster controllers. Represents first or only input or output component of a device.

Position: Must follow LINE or SERVICE definition statement. If part of a cluster, must follow the CLUSTER definition statement for the cluster.

Keywords: Table 2-117 shows the keywords described under this definition statement.

Table 2-117 (Page 1 of 2). Keywords on the TERMINAL Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ADDR	2-451				
ATTN	2-452		G,L	DISABLED	
BFRDLAY	2-452		G,L,C		
BHEXEC	2-453		G,L	ALL	
BHSET	2-454		G,L	NONE	
CONV	2-454		G,L	NO	
CRDLAY	2-455		G,L	NO	
CRITSIT	2-455		G,L,C	NO	
CTERM	2-456			NO	
CUIDLEN	2-457		G,L	0	
DIALNO	2-458				
DIALSET	2-459		G,L		
DIRECTN	2-460		G,L,C		
ENDTRNS	2-460		G,L	EOT	
EXEC	2-460		G,L,C	YES	
FANOUT	2-461				
FEATURE	2-461		G,L,C		
IDSEQ	2-463		G,L	NONE	
INHIBIT	2-464		G,L,C	NONE	
ITBMODE	2-465		G,L,C		
LCST	2-465		G,L	NONE	
LGRAPHS	2-466		G,L,C	(REJECT,REJECT)	
LMODADR	2-466		C		
NEXT	2-466				

Sift:

C   CLUSTER definition statement
G   GROUP definition statement
L   LINE definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Table 2-117 (Page 2 of 2). Keywords on the TERMINAL Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
NPACOLL	2-468			NO	
POLL	2-468				
PT3EXEC	2-469		G,L,C	NO	
SEGMENT	2-469		C		
SESSREQ	2-469			NO	
SRT	2-470		G,L	(65535,255)	
TERM	2-470	Yes	G,L,C		
VPRINT	2-472		G,L,C	NO	
XTWXID	2-473		G,L	YES	

Sift:

C   CLUSTER definition statement
G   GROUP definition statement
L   LINE definition statement

*Tuning:

Yes   Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the
      NTuneMON V2R5 User's Guide for more information.

Specifying Lower-Level Keywords on Higher-Level Definition Statements: In
addition to the keywords described in this section, you can specify most keywords
of the COMP definition statement on the TERMINAL definition statement or on the
LINE or GROUP definition statement instead of on COMP. Table 1-3 on
page 1-12 shows which of the COMP definition statement keywords you can
specify on TERMINAL.

VTAM Keywords: The following diagrams show the VTAM keywords that can be
coded on this definition statement. These keywords provide information to the
VTAM initialization process. They are not required in the generation definition you
use to generate NCP. See the *VTAM Resource Definition Reference* for
descriptions of these keywords and the *VTAM Network Implementation Guide* for
information on the VTAM initialization process.

| VTAM keywords not listed below can be specified by coding VTM. as a prefix in
| front of the VTAM keyword.

```
►►─┬──────────────────────────────────────────────┬─►◄
   └─ASLENT=associated_LU_table_entry_name─┘
```

```
►►─┬──────────────────────────────────────┬─►◄
   └─ASLTAB=associated_LU_table_name─┘
```

```
►►─┬──────────────────────────────────────────┬─►◄
   └─DLOGMOD=default_logon_mode_entry_name─┘
```

```
                  ┌─;──────┐
►►─┬──────────────▼────────┬─►◄
   └─FEATUR2=(──┴─keyword─┴─)─┘
```

```
         ┌─ISTATUS=ACTIVE────────┐
►►───────┤                       ├──►◄
         └─ISTATUS=──┬─ACTIVE───┬─┘
                     └─INACTIVE─┘


►►────────┬──────────────────────────────────┬──►◄
          └─LOGAPPL=application_program_name──┘


►►────┬──────────────────────────────┬──►◄
      └─LOGTAB=interpret_table_name──┘


►►────┬───────────────────────────────────────┬──►◄
      └─MDLENT=model_name_table_entry_name────┘


►►────┬──────────────────────────────────┬──►◄
      └─MDLTAB=model_name_table_name─────┘


►►────┬───────────────────────────────────┬──►◄
      └─MODETAB=logon_mode_table_name─────┘


►►────┬─SPDSEL=──┬─NO──────┬──┬──►◄
                 ├─NETSRVR─┤
                 └─DIRSRVR─┘


         ┌─SSCPFM=FSS───────────┐
►►───────┤                      ├──►◄
         └─SSCPFM=──┬─FSS──────┬─┘
                    ├─USSSCS───┤
                    ├─USS3270──┤
                    ├─USS3275──┤
                    ├─USS3780──┤
                    └─USSNTO───┘


►►────┬──────────────────────────────────┬──►◄
      └─USSTAB=USS_definition_table_name──┘
```

Keyword Descriptions

symbol

►►──symbol──►◄

Use: Required when you code the TERMINAL definition statement
Default: None

Provides a resource name for the device. *symbol* can be any valid assembler lan-
guage symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to
avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process
creates. If AUTOLINE is specified, you must embed the first suboperand of the
ADDRESS keyword on the LINE definition statement in the symbol.

ADDR

```
►►──┬─────────────────────────────────────────────────┬──►◄
     └─ADDR=──┬─address────────────────────────────┬─┘
              ├─selection_chars────────────────────┤
              └─(──┬────────────────┬──┬──────────────┬──)─┘
                   └─normal_address─┘  └─,alt_address─┘
```
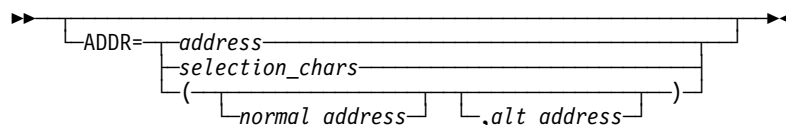
Specifies, in hexadecimal notation, the addressing or selection characters assigned to the device represented by this TERMINAL definition statement.

If you code LPDATS=LPDA2 on the LINE definition statement for this device, the first 2 characters in the value you code on ADDR cannot be FD or FF.

address

Default: None

Specifies the addressing characters for a start-stop or BSC device on a multi-point line or an IBM 1050 on a switched point-to-point line (POLLED=YES on the LINE or GROUP definition statement).

ADDR is not valid if:

- This TERMINAL definition statement represents a call-in, multiple-terminal-access, logical-connection station (TERM=MTA and CTERM=YES).

- You specify POLLED=NO on LINE and LNCTL=SS on the GROUP definition statement for the line to which this device is attached.

If this definition statement represents a call-out MTA terminal (TERM=MTA, CTERM=NO), and if any of the MTALCST definition statements named in the LCST keyword represent IBM 1050 terminals, specify either the 1050 polling or addressing characters or both for POLL and ADDR.

Code only the alphanumeric addressing characters, omitting any control characters. For example, for an IBM 2740 with station control whose address is A, code the hexadecimal representation of the EBCDIC character A (that is, ADDR=C1). Do not include the start-of-address character, which is transmitted before the addressing character, or the space character, which is transmitted following the addressing character.

The character specified for a tributary controller on a multipoint line must have the same bit pattern as the corresponding polling character defined for POLL, except that the third bit from the left (bit 2) must be 1. Bit 2 in the polling character is always 0.

selection_chars

Default: None

Specifies the component selection characters for one of the output components attached to the device on a BSC point-to-point line, if component selection is required. For IBM 2780, specify the alphanumeric component selection character but not the escape (ESC) character that precedes it. For IBM 2770, specify the device control character (DC1, DC2, or DC3) that serves as the component selection character. For all BSC devices, specify all characters except ENQ, ESC, and ETB.

normal_address
alt_address

Default: None

Specify the normal and alternate addressing characters for an IBM 2980 terminal.

Your system designer determines the addressing characters (escape sequence) assigned to the station.

A locally administered MAC address defined by ADDR on the TERMINAL definition statement cannot be the same as that defined by LOCADD on the LINE or LU definition statement.

ATTN

```
                   ┌─ATTN=DISABLED──────┐
►►─────────────────┼────────────────────┼──►◄
                   └─ATTN=─┬─DISABLED─┬──┘
                           └─ENABLED──┘
```
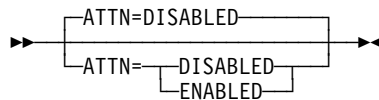
Use: IBM 1050, IBM 2741, TWX, and WTTY terminals
Default: DISABLED

Specifies whether NCP is to accept (ENABLED) or ignore (DISABLED) the attention signal of the terminal. When you specify ATTN=ENABLED, an attention signal received from the terminal causes NCP to stop sending to the terminal and notify the host processor. When you specify ATTN=DISABLED, NCP ignores the attention signal.

ATTN=ENABLED is valid only if you code FEATURE=ATTN on this TERMINAL definition statement.

BFRDLAY

```
►►─┬──────────────────┬──►◄
   └─BFRDLAY=seconds──┘
```
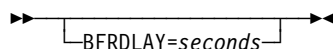
Use: Buffered terminals and nonswitched multipoint lines
Default: See text
Minimum: 0
Maximum: 255

Specifies the delay, in seconds, between successive transmissions to the device represented by this TERMINAL definition statement. BFRDLAY is valid only for the following IBM buffered devices:

IBM 2740-2 (with the buffer receive feature)
IBM 2770
IBM 3275
IBM 3284
IBM 3286
IBM 3780

If you specify TERM=3275 and a printer is attached, TERM=3284, or TERM=3286, use a buffer delay of 13 seconds or more. The 13 seconds are required to print a message that fills half the device's print buffer. Average messages that are shorter or longer require either a shorter or longer delay. BFRDLAY=0 is not valid if you code TERM=3284 or TERM=3286.

The terminal types listed return a busy response if addressed while printing. No loss of data occurs, but additional NCP overhead is required to handle the busy response and resend the message.

If you code TERM=3284 or TERM=3286, the default is BFRDLAY=13. For any other device, the default is BFRDLAY=0. However, if you code TERM=3275 and a printer is attached to the terminal, you must code BFRDLAY.

BHEXEC
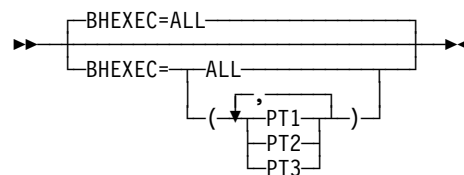
Use: Block handling routines
Default: ALL

Specifies which block handling routines NCP is to use from the block handler set defined by BHSET. Code BHEXEC only if you specify BHSET=*name* on this TER-MINAL definition statement. BHEXEC is ignored if you code BHSET=NONE or BHSET=DYNAMIC, or if you omit BHSET.

Specify at least one execution point if you code BHSET=*name*.

ALL
Specifies that each block handler in the set is to be used at the appropriate time.

PT1
Specifies that NCP is to use the PT1 block handler. NCP invokes this block handler upon receiving a contact or write request from the host processor, but before it determines whether the line is available to contact the device.

PT2
Specifies that NCP is to use the PT2 block handler. NCP invokes this block handler upon receiving a contact or write request from the host processor or after message data is received from the line. NCP invokes the block handler while the line is available for sending to or receiving from the device.

PT3
Specifies that NCP is to use the PT3 block handler. NCP invokes this block handler after receiving a block, message, or transmission from the device.

BHSET

```
                    ┌─BHSET=NONE──────┐
►►──────────────────┴─────────────────┴──────►◄
          └─BHSET=──┬─NONE────┬─┘
                    ├─DYNAMIC─┤
                    └─name────┘
```
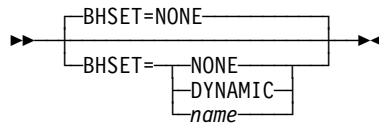
Use: Block handling routines
Default: NONE

Associates a block handler set with this device.

NONE
>   Specifies that no block handler set is to be associated with this device.

DYNAMIC
>   Specifies that no block handler set is to be associated initially with this device, but that one can be associated dynamically from the host processor.

name
>   Specifies that a block handler set is to be associated to this device. *name* must be the name of a BHSET definition statement.

For the BSC 3270, VTAM ignores BHSET.

CONV
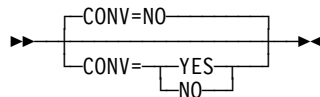
```
              ┌─CONV=NO────┐
►►────────────┴────────────┴──────►◄
      └─CONV=──┬─YES─┬─┘
               └─NO──┘
```

Default: NO

Specifies whether NCP, upon receiving a message block from a device, is to send the device a message block in response instead of a positive acknowledgment. Sending a message block in response (conversational response) is possible only if NCP currently holds a request-to-send to the device. If NCP has no request, it sends the usual positive acknowledgment.

Devices capable of accepting conversational responses are:

>   IBM 1050
>   IBM 2740 with record checking feature
>   IBM 2770 with conversational mode feature
>   All other BSC devices except IBM 2715 and 2780.

CRDLAY

```
          ┌─CRDLAY=NO──────┐
►►────────┼────────────────┼──────►◄
          └─CRDLAY=──┬─YES─┘
                     └─NO──┘
```
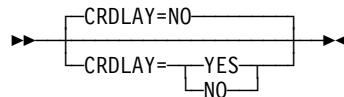
Use: Start-stop lines
Default: NO

Specifies whether NCP is to delay processing a write operation after receiving a carriage return signal. This delay does not occur if a READ command is queued in NCP as the next operation for the device.

CRDLAY=YES is valid only if this definition statement represents an IBM 1050, 2740 Model 1, 2741, TWX, or MTA terminal.

Note: No delay occurs if (1) a carriage return signal is followed by an EOT and (2) the terminal type is an IBM 1050 multipoint or is in a group of MTA terminals that includes an IBM 1050 multipoint terminal.

CRITSIT

```
          ┌─CRITSIT=NO──────┐
►►────────┼─────────────────┼──────►◄
          └─CRITSIT=──┬─YES─┘
                      └─NO──┘
```
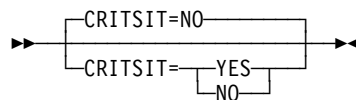
Default: NO

Specifies whether NCP is to notify this device, if the device is logically active, when NCP is about to close down the network because of a critical situation.

Note: A critical situation message cannot be sent from a tributary controller to its control device. Therefore, if NCP is running in a tributary controller and this TERMINAL definition statement defines the control device connected to the tributary controller, CRITSIT=YES is not valid. If you code TADDR on the LINE definition statement preceding this TERMINAL definition statement, do not code CRITSIT=YES.

If you code CRITSIT=YES, also code CSMSG on the BUILD definition statement to define the critical situation message.

NCP sends the notification message to the output component defined by this TERMINAL definition statement, not to any output component represented by a COMP definition statement.

If this TERMINAL definition statement defines a device on a multipoint line or an IBM 1050 on a switched point-to-point line, specify the addressing characters of the device on ADDR in order for the critical situation message to be sent.
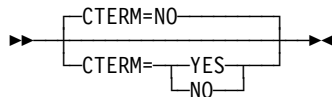
CTERM

```
      ┌─CTERM=NO─────────┐
►►─────┤                  ├────►◄
      └─CTERM=──┬─YES─┐───┘
                └─NO──┘
```

Use: Switched lines
Default: NO

Specifies whether this TERMINAL definition statement defines devices that call the controller over the switched line associated with this definition statement. The TERMINAL definition statement generates control fields to hold information about the devices. NCP uses these control fields successively for various devices that call over the line represented by the LINE definition statement preceding this TERMINAL definition statement.

Only one TERMINAL definition statement with CTERM=YES is required following each LINE definition statement on which you code CALL=IN or CALL=INOUT for a switched line. Omit CTERM for a LINE definition statement with CALL=OUT.

TERMINAL definition statements that represent devices called over switched call-out lines have no required positional relationship to the LINE definition statement for such lines. The association of devices and call-out lines in this case is made using DIALSET keywords that specify the names of logical groups. The DIALSET definition statement defines dial sets.

The TERMINAL definition statement can appear following any LINE definition statement representing a switched line whose line control matches that used by the terminals called. For example, the following definition statements specify two switched lines used for call-out only, one line used for call-in and call-out, and four devices. All three lines are included in one dial set. Five TERMINAL definition statements are required. One, in which CTERM=YES is coded, represents the logical-connection station. The remaining definition statements represent each of the four devices called over lines in the dial set.

```
DSET      DIALSET    LINES=(A,B,C)
G         GROUP      DIAL=YES,...
A         LINE       CALL=OUT,DIALSET=DSET,...
T1        TERMINAL   DIALSET=DSET,DIALNO=(5142),...
T2        TERMINAL   DIALSET=DSET,DIALNO=(7615),...
T3        TERMINAL   DIALSET=DSET,DIALNO=(8204),...
T4        TERMINAL   DIALSET=DSET,DIALNO=(7382),...
B         LINE       CALL=OUT,DIALSET=DSET,...
C         LINE       CALL=INOUT,DIALSET=DSET,...
LOGCON    TERMINAL   CTERM=YES,...
```

In this example, one or more of the TERMINAL definition statements can follow LINE definition statement B or LINE definition statement C. The resulting configuration is identical.

Note: If you code CTERM=YES, no keyword in any definition statement can refer to the name of this TERMINAL definition statement. For example, this TERMINAL definition statement cannot be named for IDSEQ on the IDLIST definition statement.

CUIDLEN

```
                    ┌─CUIDLEN=0──────────┐
►►──────────────────┴────────────────────┴──►◄
         └─CUIDLEN=──┬─length─┬──
                     ├─TWX────┤
                     └─ALL────┘
```
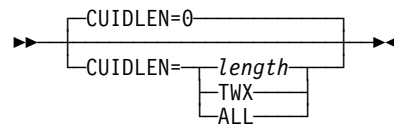
Use: Switched BSC or TWX device
Default: 0

Specifies the length of the EBCDIC identifier sequence NCP is to send to the device represented by this definition statement when the device initiates a connection.

length

> Minimum: 0
> Maximum: 20
>
> Specifies the length of the EBCDIC identifier sequence NCP is to send.
>
> If this TERMINAL definition statement defines a BSC device, *length* must be less than or equal to the length of the value of CUID on the BUILD definition statement.
>
> If this TERMINAL definition statement defines a switched TWX device, *length* must equal the length of the *inchars* suboperand of TWXID on the BUILD definition statement.

TWX
> Specifies that NCP is to send the TWX identifier to the TWX terminal defined by this TERMINAL definition statement. If you code CTERM=NO, NCP sends the entire sequence specified by the *outchars* suboperand of TWXID. If you code CTERM=YES, NCP sends the entire sequence specified by the *inchars* suboperand of TWXID.

ALL
> Specifies that NCP is to send the appropriate identifier sequence (TWXID for TWX terminals, CUID for BSC devices) to the device defined by this TERMINAL definition statement.

0
> Specifies that NCP is not to send an identifier sequence to the device defined by this TERMINAL definition statement.

CUIDLEN is valid only if DIAL=YES on GROUP definition statement and POLLED=NO on the LINE definition statement and either (1) LNCTL=SS on the GROUP definition statement and TERM=TWX on this TERMINAL definition statement or a higher-level definition statement, or (2) LNCTL=BSC on the GROUP definition statement.

DIALNO

```
►►──┬──────────────────────────────┬──►◄
    └─DIALNO=(chars──┬───────┬──)─┘
                     └─,count─┘
```

Use: Switched lines

Specifies the dial digits and the length of the telephone number NCP is to use to call this device.

chars

> Default: None
>
> Specifies the telephone number of the device. *chars* represents dial digits (EBCDIC numeric characters) NCP uses to contact the device over a switched line. You can code this parameter (1) if the device is reached by a switched connection or (2) if the switched connection is an alternate to the primary non-switched point-to-point line.
>
> For V.25 bis lines (V25BIS=(YES,...) on the LINE definition statement), the DIALNO characters contain an identifying prefix of *CRN* (call request with number) or *CRI* (call request with identification). For example, you would code DIALNO=CRN47583 for a device connected to a V.25 bis line if 47583 is the telephone number of that device.

count

> Default: None
> Minimum: 1
> Maximum: 32
>
> Specifies the number of bytes reserved to hold the telephone number.

Use *chars* or *count* as follows:

* If NCP always calls the device using the dial digits this keyword specifies, code *chars* and omit *count*, for example, DIALNO=19195678888.

* If the host processor updates the dial digits using the dynamic control facility, code *chars*. Also code *count* if the number of replacement digits can exceed the number of digits originally specified. For example, DIALNO=(19195678888,12). Otherwise, omit *count*.

DIALNO is not valid if CTERM=YES or DIALSET=NONE.

Separator and End-of-Number Characters: The sequence of dial digits can include separator or end-of-number characters if the auto-call unit at the communication controller accepts these characters. A separator character causes the auto-call unit to delay sending the next dial digit on the communication line until the auto-call unit receives a secondary dial tone. An end-of-number character sent to the auto-call unit after the last dial digit signals the auto-call unit that it has received the last digit of the telephone number and causes it to start monitoring the line for an answer tone from the distant device. Use of the end-of-number character can reduce the time needed to establish a connection, thus increasing line utilization.

You can code the separator character in the dial-digit sequence wherever a pause for a secondary dial tone is needed. Code the end-of-number character at the end of the sequence.

The communication controller sends the auto-call unit only the four low-order bits of the digits specified in the sequence. The end-of-number bit pattern is 1100 (hexadecimal C). Use any EBCDIC characters whose four low-order bits equal this pattern. The separator character, if needed, must be an underline character (or hexadecimal 6D).

Programmed Dialing Pause: If the auto-call unit is not equipped to use separator characters, you can introduce a programmed dialing pause into the dialing action to allow time to receive a secondary dial tone. At the point in the sequence where the pause is required, code one vertical bar character (X'FA') for each second of the pause. For instance, code three such characters if a 3-second pause is required: `DIALNO=(8||5799)`.

DIALSET

```
▶▶──┬─────────────────────────┬──▶◀
    └─DIALSET=──┬─name─┬──────┘
               └─NONE─┘
```
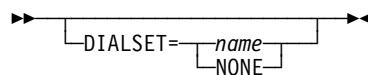
Use: Callout switched lines
Default: None

Specifies the name of the dial set (DIALSET definition statement) for this device. DIALSET is required if NCP can call the device; DIALSET is not required if NCP cannot call the device.

Each time NCP receives a call-out request for the device, it tries to call that device over one of the switched lines in the specified dial set.

Use DIALSET as follows:

- If you want NCP to call the device using the same dial set specified by the LINE definition statement preceding this TERMINAL definition statement, omit DIALSET in this definition statement. DIALSET on the LINE definition statement specifies the dial set used.

- If you want NCP to call the device using a dial set different from the one specified by the LINE definition statement preceding this TERMINAL definition statement, specify the desired dial set in DIALSET on this TERMINAL definition statement.

- If you do not need a dial set associated with this device, code DIALSET=NONE. DIALSET=NONE is appropriate when the device can call the communication controller but NCP cannot call the device.

The lines in the dial set specified by DIALSET must have the same operating characteristics as those defined by the LINE definition statement that precedes this TERMINAL definition statement.

DIRECTN

```
►►─────┬──────────────────────────────┬───►◄
       └─DIRECTN=─┬──IN────┬───────────┘
                  ├──OUT───┤
                  └──INOUT─┘
```
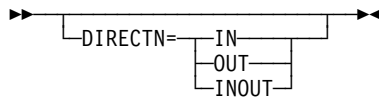
Default: See text

Specifies whether NCP will receive text from this station (DIRECTN=IN), only send text (DIRECTN=OUT), or both send and receive text (DIRECTN=INOUT).

If you omit this keyword, NCP uses the following defaults:

- If you code POLLED=YES on the LINE definition statement and specify polling characters on the POLL keyword on this definition statement, the default is DIRECTN=IN.

- If you specify addressing characters using the ADDR keyword on this definition statement, the default is DIRECTN=OUT.

- If you specify both polling and addressing characters, the default is DIRECTN=INOUT.

- If you code POLLED=NO (or omit the POLLED keyword) on the GROUP or LINE definition statement, the default is DIRECTN=INOUT.

For IBM 3284 and IBM 3286 printers, specify DIRECTN=INOUT because these devices transmit only status information (not message data) when polled.

ENDTRNS

```
        ┌─ENDTRNS=EOT──────────┐
►►──────┴──────────────────────┴──►◄
        └─ENDTRNS=─┬──EOT─┬─────┘
                   └──EOB─┘
```
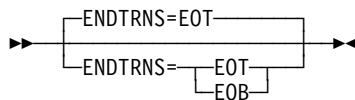
Use: Start-stop lines
Default: EOT

Specifies whether NCP is to recognize an EOT or an EOB character as the end of a transmission.

ENDTRNS is valid only for start-stop terminals that can transmit either an EOB or an EOT ending sequence. ENDTRNS is not valid for an IBM 2740 Model 2.

EXEC

```
        ┌─EXEC=YES──────────┐
►►──────┴───────────────────┴──►◄
        └─EXEC=─┬──YES─┬─────┘
                └──NO──┘
```
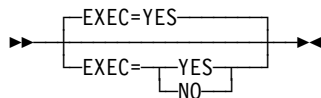
Default: YES

Specifies how the block handler is to be activated. If EXEC=YES, the block handler is activated during NCP initialization. If EXEC=NO, the block handler must be activated by a request from the host.

FANOUT

```
►►─┬─────────────────┬─►◄
   └─FANOUT=name─────┘
```
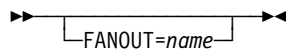
Use: BSC terminals
Default: None

Specifies that this TERMINAL definition statement is the first of a sequence of TER-MINAL definition statements representing devices attached to the same modem, and specifies the name of the last TERMINAL definition statement in the sequence. (The modem is called a fanout modem because several devices can be attached to it. The fanout modem is the modem at the distant end of the communication line, not the modem attached to the communication controller.)

For example, assume that four terminals represented by four TERMINAL definition statements named T1, T2, T3, and T4 are attached to a fanout modem. In FANOUT on the first TERMINAL definition statement, specify the name of the last TERMINAL definition statement representing a terminal attached to the same modem this way:

```
T1      TERMINAL FANOUT=(T4),....
T2      TERMINAL....
T3      TERMINAL....
T4      TERMINAL....
```

FANOUT appears only in the first TERMINAL definition statement, as shown. If only one terminal is attached to the fanout modem, omit FANOUT.

All TERMINAL definition statements representing devices attached to the same modem must appear in a single sequence, with no other intervening TERMINAL definition statements for other devices.

FEATURE

```
►►─┬───────────────────────────────────────┬─►◄
   └─FEATURE=(─┤ Terminal Features ├─(1)─)─┘
```

Terminal Features:

```
├─┬─NOACR──┬─,─┬─NOATTN─┬─,─┬─NOBREAK─┬─,─┬─NOCHECK─┬─,─┬─NOSCTL──┬─,──►
  ├─ACR───┤   ├─ATTN───┤   ├─BREAK───┤   ├─CHECK───┤   ├─SCTL────┤
  └─NOACR──┘   └─NOATTN─┘   └─NOBREAK─┘   └─NOCHECK─┘   └─NOSCTL──┘

►─┬─NOTOSUP─┬─,─┬─NOXCTL─┬─┤
  ├─TOSUPPR─┤   ├─XCTL───┤
  └─NOTOSUP─┘   └─NOXCTL─┘
```
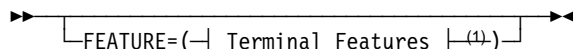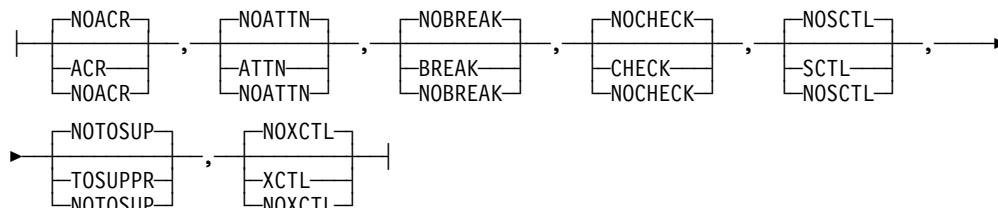
[1] Order of Terminal Features does not matter.

Use: Start-stop lines in network control mode
Example: FEATURE=(ATTN,CHECK,XCTL)

Specifies the machine features with which this terminals is equipped. You can code the suboperands in any order. Separate the suboperands by commas.

ACR│NOACR

Default: NOACR

Specifies whether the device is equipped with the accelerated carrier return feature. NCP makes use of the accelerated carrier return feature only if you specify FEATURE=ACR for all TERMINAL definition statements following the LINE definition statement. If you omit FEATURE=ACR in any TERMINAL definition statement, NCP ignores the presence of the feature when sending to any device on the line. For an IBM 1050 terminal accessed through MTA, refer to ACR on the MTALCST definition statement.

ATTN│NOATTN

Default: NOATTN

Specifies whether the terminal can send attention signals to the controller. For IBM 1050 or 2741 terminals, specify FEATURE=ATTN only if the terminal is equipped with the receive interrupt (1050) or interrupt (2741) feature.

BREAK│NOBREAK

Default: NOBREAK

Specifies whether NCP can interrupt a transmission from the device by sending a break signal. For IBM 1050 or 2741 terminals, code FEATURE=BREAK only if the terminal is equipped with the transmit interrupt feature. FEATURE=BREAK is valid only if you specify DUPLEX=FULL or SCLSET=YES on the LINE definition statement and XBREAK=*integer* on the BUILD definition statement. An exception is that for an IBM 3767 specified as an IBM 2741 (TERM keyword on the TERMINAL definition statement), FEATURE=BREAK also is valid if you specify DUPLEX=HALF.

CHECK│NOCHECK

Default: NOCHECK

Specifies whether the device is equipped with the record checking feature.

SCTL│NOSCTL

Default: NOSCTL

Specifies whether the device is equipped with the station control feature. (This feature is optional for the IBM 2740 Model 1 but standard for the Model 2.)

TOSUPPR│NOTOSUP

Default: NOTOSUP

Specifies whether the terminal is equipped with the time-out suppression feature.

XCTL│**NOXCTL**

> Default: NOXCTL

> Specifies whether the terminal is equipped with the transmit control feature.

IDSEQ
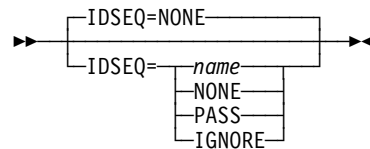
Use: IBM 2740 Model 1 and BSC and TWX devices on switched lines
Default: NONE

Specifies whether the devices defined by this TERMINAL definition statement can transmit identification sequences when calling or being called by the controller, and if so, how NCP is to react to the sequence.

IDSEQ is valid only if both of the following are true:

* The device is a BSC or TWX device on a switched line (LNCTL=BSC or LNCTL=SS and DIAL=YES on the GROUP definition statement; the line was not defined with POLLED=YES; and TERM on this TERMINAL definition statement specifies a BSC or TWX device on TERM.

* The controller can expect an identifier sequence from a device it calls (this TERMINAL definition statement specifies a telephone number on DIALNO).

name
> Specifies that NCP is to expect and verify identifier sequences received from devices. *name* also names the identification list (IDLIST definition statement) NCP is to use to check the sequences it receives.

> *name* is valid only if the device defined by this TERMINAL definition statement is not a multiple-terminal-access station (that is, the value of TERM is not MTA). *name* is valid whether this TERMINAL definition statement defines a call-in logical-connection station (CTERM=YES) or a call-out station (CTERM=NO).

NONE
> Specifies that no identifier sequences are expected from devices that call the controller or are called by the controller.

PASS
> Specifies that NCP is to expect identifier sequences from devices and pass to VTAM all identifier sequences it receives. IDSEQ=PASS is valid only if this TERMINAL definition statement defines a call-in logical-connection station (CTERM=YES).

IGNORE
> Specifies that NCP is to ignore—that is, neither check nor pass to the host processor—any identifier sequences it receives. IDSEQ=IGNORE is valid only if the controller calls the device represented by this TERMINAL definition statement. If devices call the controller (CTERM=YES), IDSEQ=IGNORE is not valid.

INHIBIT
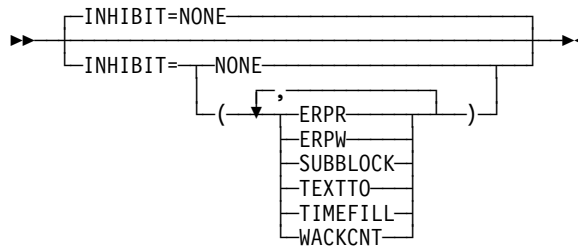
```
                ┌─INHIBIT=NONE──────────────────────────────┐
  ►►────────────┼───────────────────────────────────────────┼────►◄
                └─INHIBIT=─┬─NONE─────────────────────────┐
                           │          ┌─────,─────┐        │
                           └─(──▼─────┬─ERPR─┬─────)──────┘
                                      ├─ERPW─────┤
                                      ├─SUBBLOCK─┤
                                      ├─TEXTTO───┤
                                      ├─TIMEFILL─┤
                                      └─WACKCNT──┘
```

Default: NONE
Example: INHIBIT=(SUBBLOCK,ERPW)

Specifies which NCP facilities are to be inhibited from functioning initially, that is, when NCP begins running after being loaded into the controller. You can code the suboperands in any order. Separate the suboperands by commas.

Note: For BSC 3270 lines, it is recommended that you code INHIBIT=SUBBLOCK and CUTOFF=1 on the LINE definition statement.

NONE
Specifies that no NCP facilities are to be inhibited. All NCP facilities function when the program starts to run.

TEXTTO
Specifies that NCP is to use the interval coded on ITEXTTO on the BUILD definition statement, rather than the value defined by TEXTTO on the GROUP definition statement, as the time limit between receipt of successive text characters.

TIMEFILL
Specifies that automatic insertion of idle characters following carriage return and horizontal tab characters is to be inhibited. No idle characters are sent. TIMEFILL is valid only if you code LNCTL=SS or do not code LNCTL on the GROUP definition statement.

WACKCNT
Specifies that the WACK limit specified by WACKCNT on the GROUP definition statement is to be inhibited. The first WACK received from the device causes NCP to return the request to the host processor rather than to respond to that and subsequent WACKs with an ENQ character. WACKCNT is valid only if you code LNCTL=BSC on the GROUP definition statement.

SUBBLOCK
Specifies that the subblocking indicated by TRANSFR on the LINE or GROUP definition statement is to be inhibited. If the number of buffers specified by TRANSFR is filled by received text, NCP terminates the receiving operation as if the cutoff limit, specified by the CUTOFF keyword, had been reached.

If you do not inhibit subblocking, you must code a value of 2 or more for TRANSFR.

ERPR
Specifies that recovery procedures for text-read errors, as defined by RETRIES on the LINE definition statement, are to be inhibited. NCP does not attempt recovery for text-read errors.

ERPW

Specifies that recovery procedures for text-write errors, as defined by RETRIES on the LINE definition statement, are to be inhibited. NCP does not attempt recovery for text-write errors.

ITBMODE

```
►►─┬─────────────────────────────────────┬─►◄
   └─ITBMODE=(─┬──────┬─,─┬──────┬─)─┘
              ├─YES─┤     ├─YES─┤
              └─NO──┘     └─NO──┘
```
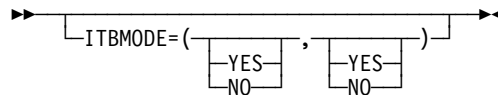
Use: BSC devices in ITB mode
Default: None

Specifies how NCP is to handle intermediate text block (ITB) characters in text received from the station or the access method.

The first suboperand specifies whether NCP is to insert an error information block (EIB) character following each ITB character received from the station.

The second suboperand specifies whether an EIB character follows each ITB character received from the access method. If you specify YES, NCP removes the first character following each ITB character from the data before sending the data to the station.

LCST

```
         ┌─LCST=NONE──────┐
►►─┬──────────────────────┬─►◄
   └─LCST=─┬─name─┬───┘
           └─NONE─┘
```
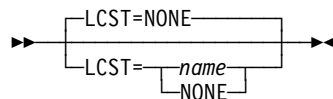
Use: MTA lines
Default: NONE

Specifies the name of the MTALCST definition statement that defines the device characteristics NCP uses when calling the device defined by this TERMINAL definition statement.

Code LCST=*name* on the TERMINAL definition statement for each device that NCP calls using a multiple-terminal-access line.

Code LCST=*name* only if you:

- Specify DIAL=YES and either omit LNCTL or specify LNCTL=SS on the GROUP definition statement
- Specify TERM=MTA on the TERMINAL, LINE, or GROUP definition statement
- Omit CTERM or specify CTERM=NO on the TERMINAL definition statement
- Specify a telephone number for DIALNO on the TERMINAL definition statement
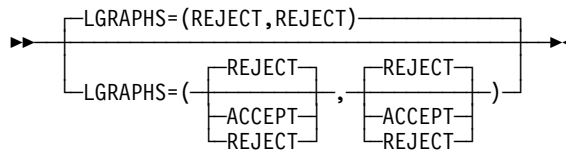
LGRAPHS

```
        ┌─LGRAPHS=(REJECT,REJECT)──────────────┐
►►──────┤                                      ├──►◄
        │           ┌─REJECT─┐     ┌─REJECT─┐  │
        └─LGRAPHS=(─┼────────┼──,──┼────────┼─)┘
                    ├─ACCEPT─┤     ├─ACCEPT─┤
                    └─REJECT─┘     └─REJECT─┘
```

Default: REJECT

Specifies whether NCP is to accept or reject leading graphics received from the station.

The first suboperand applies to read operations. The second suboperand applies to write operations.

IBM 2740 Model 2: The second suboperand must be ACCEPT.

LMODADR

```
►►──┬───────────────────────────────┬──►◄
    └─LMODADR=(address1─┬──────────┬─)┘
                        └─,address2─┘
```
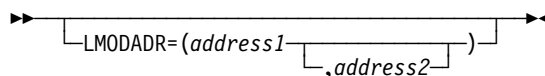
Specifies the local modem addresses used on the line going to the terminal. LMODADR is valid on the TERMINAL statement only if you specify LNCTL=BSC on the GROUP definition statement and LPDATS=LPDA2 on the LINE definition statement. The values of *address1* and *address2* must be hexadecimal values ranging from X'0' to X'F'. *address1* specifies the local modem address on link segment 1; *address2* specifies the local modem address on link segment 2. *address2* is valid only if SEGMENT=2. The default value is LMODADR=01 if SEGMENT=1 and LMODADR=(01,02) if SEGMENT=2.

If the TERMINAL definition statement is defined under a CLUSTER definition statement, and LMODADR is coded on both definition statements, the values of LMODADR must be equal.

NEXT

```
►►──┬────────────────────────────┬──►◄
    │                  ┌─D─┐      │
    └─NEXT=(nextlabel,─┼───┼─)────┘
                       ├─D─┤
                       └─H─┘
```

Specifies a convention for constructing unique labels for the definition statements that will be copied when you code the AUTOCOPY keyword on the LINE definition statement. NEXT is valid only when you code AUTOCOPY on the previous LINE definition statement. NEXT is required when you code AUTOCOPY on the LINE definition statement.

See AUTOCOPY on the LINE definition statement on page 2-196 for more information. See the *NCP, SSP, and EP Resource Definition Guide* for more information about using NEXT.

nextlabel

Default: None

Specifies the label for the first duplicated copy of this TERMINAL definition statement. *nextlabel* is required when you code AUTOCOPY on the LINE definition statement. The label consists of 1 to 8 characters; the first character must be an alphabetical character from A to Z, #, or @. NDF uses this *nextlabel* to determine what positions in the label to use for incrementing label names and the amount of the increment. The value you enter for *nextlabel* must be the same length as the statement label, and must differ from the label in such a way that NDF can identify a numeric field to use for incrementing. You can use only one field for incrementing. This field can increase in length as it is incremented, but it cannot increase so much that the total label length becomes greater than 8 characters.

The format of the statement *label* is

`<p>lo_num_field<s>`

The format of *nextlabel* is

`<p>next_num_field<s>`

where:

- `label` is the name specified on this TERMINAL definition statement

- `nextlabel` is the label given to the first copy of this TERMINAL definition statement

- `p` is the prefix of the numeric field and must be the same for both labels

- `s` is the suffix of the numeric field and must be the same for both labels

NDF subtracts `lo_num_field` from `next_num_field` to obtain the value it uses to increment the numeric field for the next label.

The numeric fields cannot differ in length, but must differ in value. `next_num_field` must be greater than `lo_num_field`.

If NDF cannot determine how to increment a label, it issues an error message.

D|**H**

Default: D

Specifies whether the increment and the field to increment are decimal (D) or hexadecimal (H). What you specify here does not need to match what you code for AUTOCOPY or for NEXT on other resource definition statements to be replicated.
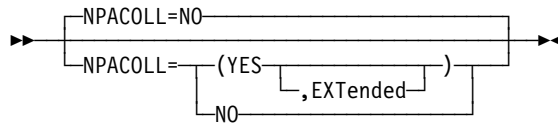
NPACOLL

```
                    ┌─NPACOLL=NO─────────────────────────────┐
►►──────────────────┴────────────────────────────────────────┴──►◄
                    └─NPACOLL=─┬─(YES─────────────────)─┬─
                               │         └─,EXTended─┘   │
                               └─NO─────────────────────
```

Default: NO

Specifies whether performance data for this terminal is eligible for collection by the network performance analyzer. NPACOLL is valid only if NPACOLL=YES on the CLUSTER definition statement and TERM=3275, 3277, 3284, or 3286 on this definition statement.

EXTENDED
EXTENDED indicates that the form of the data collected by the network performance analyzer is different from the form used by NCP.

POLL
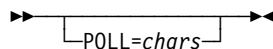
```
►►──┬──────────────┬──►◄
    └─POLL=chars─┘
```

Specifies the hexadecimal representation of the EBCDIC polling characters assigned to this device.

If you code LPDATS=LPDA2 on the LINE definition statement for this device, the first 2 hexadecimal digits in the value you specify on POLL cannot be FD or FF.

Code only the alphanumeric polling character, omitting any control characters. For example, for an IBM 2740 with station control whose polling character is A, code the hexadecimal representation of the EBCDIC character A (that is, POLL=C1). Do not code the space character that is transmitted after the polling character.

If you code TERM=3275, 3277, 3284, or 3286 and ADDR specifies the addressing character, POLL defines the polling character.

If this TERMINAL definition statement represents a call-in, logical-connection IBM 1050 station (CTERM=YES, TERM=1050), specify polling characters on at least one of the definition statements representing this station. You may specify these polling characters on this TERMINAL definition statement or on one of the COMP definition statements, if any, representing station components.

POLL is not valid if this TERMINAL definition statement represents:

* An IBM 2740 without station control, an IBM 2741, or an IBM 2980, because these types of terminals cannot be polled individually

* A call-in, multiple-terminal-access, logical-connection station (TERM=MTA, CTERM=YES)

If this TERMINAL definition statement represents an MTA station (TERM=MTA) that is not a call-in MTA station (CTERM=NO), you may specify or omit POLL or ADDR. An exception is that if any of the MTALCST definition statements named in LCST on this TERMINAL definition statement represent IBM 1050 terminals, you must define either POLL or ADDR, or both, in this TERMINAL definition statement.

The character you code on POLL for a tributary controller must conform to the requirements described under TADDR on the LINE definition statement.

PT3EXEC
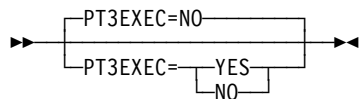
```
              ┌─PT3EXEC=NO──────────┐
►►────────────┤                     ├────────────►◄
              └─PT3EXEC=──┬─YES─┬───┘
                          └─NO──┘
```

Use: Block handling routines
Default: NO

Specifies whether a block handler set that runs at point 3 is associated with the device represented by this TERMINAL definition statement. See the BHSET definition statement for more information.

PT3EXEC is ignored if you omit BHSET or you code BHEXEC=NONE, BHEXEC=PT3, or BHEXEC=ALL on this TERMINAL definition statement.

SEGMENT

```
►►──┬──────────────────────┬──►◄
    └─SEGMENT=──┬─1─┬───────┘
                └─2─┘
```
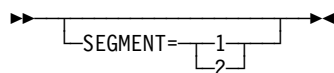
Specifies which link segment a terminal is on, which allows NCP to set up LPDA2 tests accordingly. SEGMENT is valid only if you specify LPDATS=LPDA2 on the LINE definition statement. SEGMENT=1 indicates the station is on terminal number 1; SEGMENT=2 indicates the terminal is on segment number 2. SEGMENT=1 is valid only if you specify TAILING=NO on the LINE definition statement, and SEGMENT=2 is valid only if you specify TAILING=YES.

If the TERMINAL definition statement is defined under a CLUSTER definition statement, and SEGMENT is defined on both, the values must be equal.

SESSREQ

```
              ┌─SESSREQ=NO──────────┐
►►────────────┤                     ├────────────►◄
              └─SESSREQ=──┬─YES─┬───┘
                          └─NO──┘
```
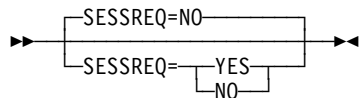
Use: IBM 3270 on a BSC line
Default: NO

Specifies whether a session with a terminal must exist before NCP can start polling for data. This prevents the loss of data sent when no session is established between the sending BSC device and the host.
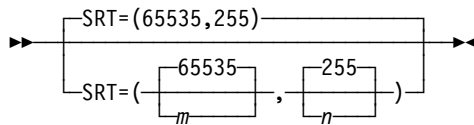
SRT

```
                      ┌─SRT=(65535,255)────────────┐
►►─────────────────────┴────────────────────────────┴──►◄
          └─SRT=(─┬─65535─┬─,─┬─255─┬─)─┘
                  └──m────┘   └──n──┘
```

Specifies a threshold value for the total number of transmissions and for the total number of error retries associated with this terminal. When either of these thresholds is reached, NCP generates a RECMS or problem determination statistics (PDstats) NMVT PIU to inform the host of the condition.

m

Default: 65535
Minimum: 1
Maximum: 65535

Specifies the value of the traffic count threshold.

Attention: A transmission threshold value of 1 is not recommended because LPDA testing can cause other processing to be locked out.

n

Default: 255
Minimum: 1
Maximum: 255

Specifies the value of the error count threshold.

SRT is not valid if you specify LEVEL5=USER on the GROUP definition statement. For IBM 786x modems extended LPDA2 capability lines connected to port B, C, or D, code the SRT thresholds to a high value or allow SRT to default to the maximum value. This reduces the performance degradation of the port A line while LPDA2 tests are being run on the line on port A to generate statistical records for lines on ports B, C, and D.

TERM

```
►►───TERM=type───►◄
```

Use: Required when you code the TERMINAL definition statement
Default: None

Specifies the type of device defined by this TERMINAL definition statement. The device type must be one of those listed in Table 2-118 on page 2-471.

If the line the device is attached to operates in emulation mode and in network control mode, specify TERM on the LINE or GROUP definition statement and on the TERMINAL definition statement.

If this TERMINAL definition statement defines a call-in logical-connection station (CTERM=YES) representing more than one type of BSC device, code any of the types in this keyword. For example, if this logical-connection station represents IBM 1130, 2780, and System/370 Model 135 devices, you can specify TERM=1130, TERM=2780, or TERM=3135.

Define the device on this TERMINAL definition statement or on either the LINE or GROUP definition statement.

If you code SPEED on the LINE definition statement as either AUTOICC1 or AUTOICC2, you must code TERM=TWX.

Table 2-118 (Page 1 of 2). Values for TERM on the TERMINAL Definition Statement

If Type of Device Is	Code TERM=
IBM 1050 Data Communication System	1050
IBM 1130 Computing System	1130
IBM 1800 Data Acquisition and Control System	1800
IBM System/360 Model 20	2020
IBM System/360 Model 25	2025
IBM 2701 Data Adapter Unit	2701
IBM 2715 Communications Terminal Model 1	2715
IBM 2740 Transmission Control Unit Model 2	2740-1
IBM 2740 Communications Terminal Model 2	2740-2
IBM 2741 Communications Terminal	2741
IBM 2770 Data Communications System	2770
IBM 2780 Data Transmission Terminal	2780
One of the following IBM 3270 Information Display Systems:	
IBM 3275 Display Station[1]	3275
IBM 3276 Control Unit Display Station	3277
IBM 3277 Display Station[1]	3277
IBM 3278 Display Station	3277
IBM 3284 Printer[1]	3284
IBM 3286 Printer[1]	3286
IBM 3287 Printer	3284
IBM 3288 Line Printer	3286
IBM 3289 Line Printer	3286
IBM 3650 Retail Store System (in BSC mode)	SYS3
IBM 3660 Supermarket System (in BSC mode)	SYS3
IBM 3704 Communications Controller	3704
IBM 3735 Programmable Buffered Terminal	3735
One of the following IBM 3740 Data Entry Systems:	
IBM 3741 Data Station	3741
IBM 3747 Data Converter	3747

[1] Not supported on switched lines.
[2] Specify a System/7 (BSC version) as follows:
- Specify TERM=SYS3 on the TERMINAL, LINE, or GROUP definition statement if the line operates in network control mode.
- Specify TERM=SYS3 on the LINE or GROUP definition statement if the line operates in both network control and emulation modes (alternately).
- Specify TERM=SYS7 on the LINE or GROUP definition statement if the line operates in emulation mode.

Table 2-118 (Page 2 of 2). Values for TERM on the TERMINAL Definition Statement

If Type of Device Is	Code TERM=
IBM 3767 Communications Terminal (in start-stop mode) supported as one of the following:	
2740 Model 1	2740-1
2740 Model 2	2740-2
2741	2741
IBM 3770 Data Communications System (in BSC mode)	2770
IBM 3780 Data Communications Terminal	3780
IBM System/370 Model 125	3125
IBM System/370 Model 135	3135
IBM System/3	SYS3
IBM System/7 (BSC version)[1]	[2]
IBM System/7 (SS version)	2740-1
IBM System/32 (BSC version)	SYS3
IBM Communicating Magnetic Card Selectric Typewriter	2741
Western Union Teletypewriter Exchange Service	TWX
World Trade Teletypewriter Terminals[1]	WTTY
Multiple Terminal Access (IBM 1050,2740,2741;TWX)[1]	MTA

[1]   Not supported on switched lines.

[2]   Specify a System/7 (BSC version) as follows:
- Specify TERM=SYS3 on the TERMINAL, LINE, or GROUP definition statement if the line operates in network control mode.
- Specify TERM=SYS3 on the LINE or GROUP definition statement if the line operates in both network control and emulation modes (alternately).
- Specify TERM=SYS7 on the LINE or GROUP definition statement if the line operates in emulation mode.

VPRINT

```
            ┌─VPRINT=NO──────┐
►►──────────┼────────────────┼──────►◄
            └─VPRINT=──┬─YES─┘
                       └─NO──┘
```
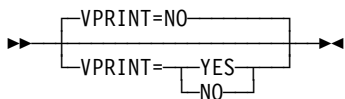
Default: NO

Specifies whether NCP is to treat this device as a virtual printer. Specify VPRINT=YES for devices that require special printer processing but may not be true printers. For example, specify VPRINT=YES for graphic devices that are defined with TERM=3286 or TERM=3284 on the LINE definition statement.

VPRINT=YES is valid only if you specify BFRDLAY and code a value other than zero.

XTWXID

```
               ┌─XTWXID=YES─────┐
►►─────────────┴────────────────┴──►◄
        └─XTWXID=──┬─YES─┬─
                   └─NO──┘
```
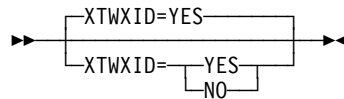
Use: TWX switched lines
Default: YES

Specifies whether, at line activation time, NCP is to transmit the identifier defined
by TWXID on the BUILD definition statement.

The TWXID sequence is referred to as an *ID* at line activation time and a *PROMPT*
at READ time.

UBHR Definition Statement

Usage: User-written block handling routines for BSC or start-stop devices. There is no limit on number allowed.

Note: Use of UBHR forces storage boundary alignment to the next 2KB boundary because of the storage-protect feature of the communication controller. Therefore, up to 2KB of storage can be unused when NCP is loaded into the controller.

Position: Can appear in any sequence between the STARTBH and ENDBH definition statement.

Keywords: Table 2-119 shows the keywords described under this definition statement.

Table 2-119. Keywords on the UBHR Definition Statement

Keyword	Page	Required	Sift	Default	Tuning*
ACCESS	2-475			GOOD	
COMMAND	2-475				
ENTRY	2-475				
NAME	2-475	Yes			
PT2EXEC	2-476			BEFORE	

*Tuning:

Yes Tuning can be done using NTuneMON with its tuning feature NTuneNCP. See the *NTuneMON V2R5 User's Guide* for more information.

Keyword Descriptions

symbol

Default: None

Provides a name for the definition statement. *symbol* can be any valid assembler language symbol that does not duplicate any existing symbol.

See Table 1-4 and Table 1-5 on page 1-22 for reserved prefixes and labels to avoid when naming resources.

The symbol appears in the label cross-reference listing that the generation process creates.

ACCESS

```
                 ┌─ACCESS=GOOD─────────────┐
►►───────────────┤                         ├──►◄
                 └─ACCESS=──┬─GOOD──┬───────┘
                           └─ERROR─┘
```
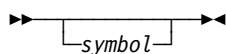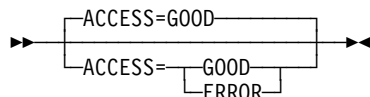
Default: GOOD

Specifies whether the user block handling routine is to process only error-free blocks (ACCESS=GOOD) or blocks containing errors (ACCESS=ERROR).

Code ACCESS only if the UBHR definition statement follows a STARTBH definition statement on which you specify BHEXEC=PT2 or BHEXEC=PT3. If you define BHEXEC=PT2, also code PT2EXEC=AFTER on the UBHR definition statement. If this UBHR definition statement follows a STARTBH definition statement that speci-fies BHEXEC=PT1, ACCESS is not valid and must be omitted.

COMMAND
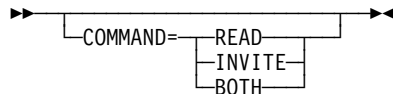
```
►►──────┬────────────────────────┬──►◄
        └─COMMAND=──┬─READ───┬────┘
                    ├─INVITE─┤
                    └─BOTH───┘
```

Specifies whether this user-written routine processes data received in response to a Read request, an Invite request, or both. This specification applies only to incoming data when the UBHR definition statement follows a STARTBH definition statement that specifies BHEXEC=PT2 or BHEXEC=PT3.

COMMAND is valid only if you omit ACCESS or code ACCESS=GOOD on this definition statement.

If this UBHR definition statement follows a STARTBH definition statement that specifies BHEXEC=PT1, COMMAND is not valid.

ENTRY

```
►►──────┬─────────────────┬──►◄
        └─ENTRY=name──────┘
```
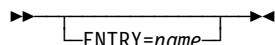
Specifies the name of the entry point within the user-written module identified by the NAME keyword.

If you omit ENTRY, the entry-point name is assumed to be the same as the module name.

NAME
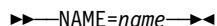
```
►►───NAME=name───►◄
```

Use: Required when you code the UBHR definition statement
Default: None

Specifies the name of the user-written module that contains the block handling routine defined by this UBHR definition statement. The module name must be con-tained in the data set specified by SNCPMOD1. For MVS, this data set can also be concatenated to SNCPMOD1.

PT2EXEC

```
                    ┌─PT2EXEC=BEFORE────────┐
   ►►───────────────┴───────────────────────┴──►◄
              └─PT2EXEC=─┬─BEFORE─┬─┘
                        └─AFTER──┘
```
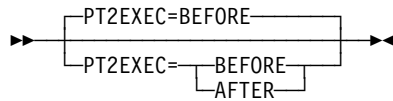
Default: BEFORE

Specifies whether the user-written routine processes outgoing message data before or after the I/O operation.

PT2EXEC is valid only on a UBHR definition statement that follows a STARTBH definition statement that specifies BHEXEC=PT2.

Appendix A. Keyword Summary

This appendix contains a brief description of each NCP keyword. The keywords appear in alphabetical order. The description of each keyword includes:

- Its syntax
- A brief description of its function
- A list of the definition statements on which it can be coded
- The number of the page in this book where it is described in detail

If a keyword has a different function or syntax on different definition statements, a separate keyword description appears for each different function or syntax.

See "Keyword Syntax Diagrams" on page 1-2 for a description of the conventions used in the diagrams.

A-1

ADDR

Keyword	Description

ACCESS

```
    ┌─ACCESS=GOOD─────┐
►►──┤                 ├──►◄
    └─ACCESS=─┬─GOOD──┘
             └─ERROR─┘
```
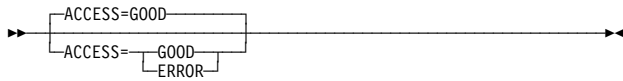
Specifies whether the user block handling routine is to process only error-free blocks or blocks containing errors.

Code on: UBHR (page 2-475).

ACR

```
    ┌─ACR=NO────────┐
►►──┤               ├──►◄
    └─ACR=─┬─YES──┘
          └─NO──┘
```

Specifies whether the IBM 1050 Data Communication System terminals that can call the communication controller on this line are equipped with the accelerated carriage return feature.

Code on: MTALCST (page 2-317).

ACTIVTO

```
►►──┬──────────────────┬──►◄
    └─ACTIVTO=seconds──┘
```

For secondary mode. Specifies how long the secondary NCP is to wait for communication from the primary NCP before initiating shutdown.

Code on: GROUP (page 2-105).

ACTPU

```
    ┌─ACTPU=YES─────┐
►►──┤               ├──►◄
    └─ACTPU=─┬─YES──┘
            └─NO──┘
```
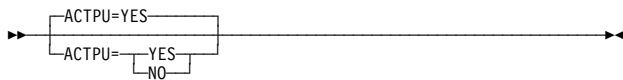
Specifies whether access methods in this non-native network can activate the gateway NCP.

Code on: NETWORK (page 2-332).

ADAPTER

```
►►──┬─────────────────────┬──►◄
    └─ADAPTER=─┬─TIC1─┐
              ├─TIC2─┤
              └─TIC3─┘
```

For Token-Ring physical lines. Specifies which type of Token-Ring interface coupler is attached to this group of Token-Ring physical lines.

Code on: GROUP (page 2-106).

ADAPTER

```
    ┌─ADAPTER=TIC1────────┐
►►──┤                     ├──►◄
    └─ADAPTER=─┬─TIC1─┐
              ├─TIC2─┤
              └─TIC3─┘
```

For Token-Ring physical lines. Specifies which type of Token-Ring interface coupler is attached to this Token-Ring physical line.

Code on: LINE (page 2-179).

ADDIFG

```
    ┌─ADDIFG=NO─────┐
►►──┤               ├──►◄
    └─ADDIFG=─┬─YES──┘
             └─NO──┘
```
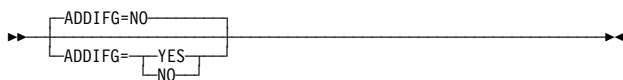
Specifies whether the gap between frames should be made larger.

Code on: LINE (page 2-180).

ADDR

```
►►──┬───────────────┬──►◄
    └─ADDR=address──┘
```

For nonswitched links. Specifies, in hexadecimal, the 8-bit address of this physical unit.

Code on: PU (page 2-366).

ADDR

```
►►──┬─ADDR=─┬─address──────────────────────────────┬──►◄
           ├─selection_chars───────────────────────┤
           └─(──┬───────────────┬─┬──────────────┬─)┘
                └─normal_address─┘ └─,alt_address─┘
```
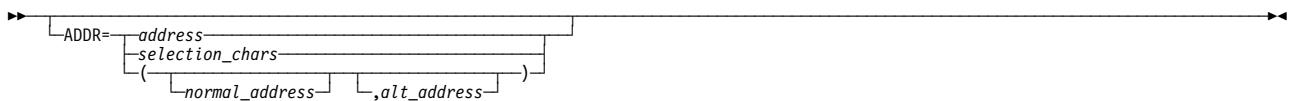
Specifies, in hexadecimal, the addressing or selection characters assigned to this device.

Code on: TERMINAL (page 2-451), COMP.

Keyword	Description

ADDRCONV

Converts the ADDR keyword on frame relay PU statements to the DLCI keyword in a format of your choice (decimal, hexadecimal, or dotted decimal).

Code on: OPTIONS (page 2-342).

ADDRESS

Relative line number:

Specifies the relative line number and data transfer mode, or emulation subchannel addresses, for this line. Also specifies the channel adapter logical address for user-written channel code, NCP subarea channel links, and NCP peripheral channel links.

Relative line number (emulation mode):

Code on: LINE (page 2-180).

ADDSESS

Specifies how many LU-LU boundary session control blocks are to be made available for use by any independent logical unit in addition to those reserved by RESSCB on the LU definition statement.

Code on: BUILD (page 2-10).

ANS

For BSC 3270 devices. Determines session continuation for BSC 3270 devices when NCP goes into ANS for an owning SSCP because the SSCP is no longer connected to NCP.

Code on: GROUP, LINE (page 2-193).

ANS

For SDLC (LNCTL=SDLC) and channel adapter stations. Specifies whether this device is to stop or continue to operate when NCP enters automatic network shutdown.

Code on: GROUP, LINE, PU (page 2-369).

ANSTONE

For call-in switched lines in network control mode. Specifies whether NCP transmits an answer tone to the calling device to signify completion of the line connection.

Code on: GROUP, LINE (page 2-193).

Keyword	Description

ARPTAB

```
          ┌─ARPTAB=(4,,NOTCANON)──────────────────────────────┐
►►────────┤                                                   ├──────────►◄
          │         ┌─4,NOTCANON─┐                            │
          └─ARPTAB=─(─┤            ├─,─,─┬──────────┬─)────────┘
                      └─entries────┘     └─NOTCANON─┘
```

For frame-relay internet devices. For IP over frame relay devices, ARPTAB specifies the maximum number of temporary entries that can exist in the ARP table at any given time.

Code on: LINE (page 2-194).

ARPTAB

```
          ┌─ARPTAB=(1000,20,NOTCANON)─────────────────────────────┐
►►────────┤                                                       ├──────►◄
          │         ┌─1000────┐  ┌─20─┐  ┌─NOTCANON─┐             │
          └─ARPTAB=─(─┤         ├─,─┤    ├─,─┼─CANON────┼─)─────────┘
                      └─entries─┘  └─t──┘  └─NOTCANON─┘
```
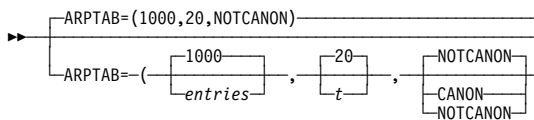
For Internet Protocol. Specifies the size of the Address Resolution Protocol (ARP) table and how NCP manages the entries in the table.

Code on: PU (page 2-371).

ATTN

```
          ┌─ATTN=DISABLED──────┐
►►────────┤                    ├──────►◄
          └─ATTN=─┬─DISABLED─┬─┘
                  └─ENABLED──┘
```
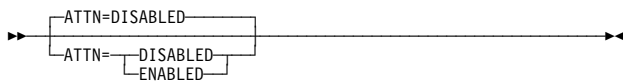
For IBM 1050, IBM 2741, TWX, and WTTY terminals. Specifies whether NCP is to accept or ignore the attention signal of this terminal.

Code on: GROUP, LINE, TERMINAL (page 2-452), COMP.

AUTO

```
►►──┬──────────────────────────────────┬──────►◄
    └─AUTO=(─┬─lnbr─┬─┬──────────┬─)────┘
             └─YES──┘ └─,digits──┘
```
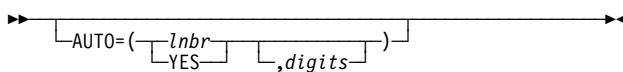
For switched lines. Specifies the relative line number of an automatic calling unit and the maximum number of dial digits allowed in a DIAL command.

Code on: LINE (page 2-194).

AUTOCOPY

```
►►──┬──────────────────────────────────────────────┬──────►◄
    │                                    ┌─D─┐      │
    └─AUTOCOPY=(copies,nextlabel,─┬───┬─,─┬──────┬─)─┘
                                  ├─D─┤  └─DONE─┘
                                  └─H─┘
```
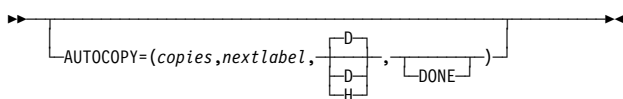
For automatic line generation. Specifies the number of copies to be made of a designated line and its devices.

Code on: LINE (page 2-196).

AUTODL

```
          ┌─AUTODL=YES─┐
►►────────┤            ├──────►◄
          └─AUTODL=─┬─YES─┬─┘
                    └─NO──┘
```
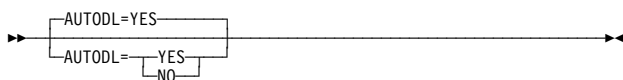
For switched X.21 lines. Specifies whether the X.21 data circuit terminating equipment accepts dial digits.

Code on: GROUP, LINE (page 2-198).

AUTOGEN

```
►►──┬──────────────────┬──────►◄
    └─AUTOGEN=number───┘
```

For Token-Ring, frame-relay, and ISDN logical switched resources. Specifies how many LINE and PU definition statement pairs for Token-Ring logical peripheral lines NDF is to add automatically to this GROUP definition statement.

Code on: GROUP (page 2-107).

Keyword	Description

AUTOLINE

```
►►──┬─────────────────────────────┬──►◄
     └─AUTOLINE=(target─┬──────┬─)─┘
                        └─,DONE─┘
```
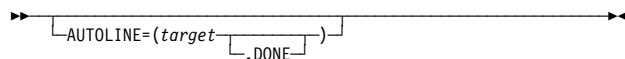
For automatic line generation. Causes NDF to create copies of this LINE definition statement and all definition statements that are subordinate to it.

Code on: LINE (page 2-199).

AUTUACB

```
►►──┬──────────────────┬──►◄
     └─AUTUACB=symbol───┘
```

For user-written line control. Specifies the name of the user adapter control block (UACB) associated with the relative line number specified by AUTO on this LINE definition statement.

Code on: GROUP, LINE (page 2-200).

AUXADDR

```
      ┌─AUXADDR=0───────┐
►►──┬─┴─────────────────┴─┬──►◄
     └─AUXADDR=count───────┘
```
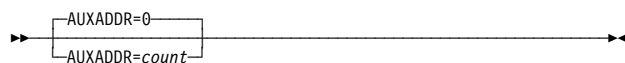
Specifies how many additional addresses can be assigned to all peripheral logical units.

Code on: BUILD (page 2-10).

AVGPB

```
►►──┬─────────────────┬──►◄
     └─AVGPB=bytes─────┘
```

For BSC and start-stop devices in network control mode. Specifies the average number of bytes expected from a device when it is polled.

Code on: GROUP, LINE (page 2-200).

AVGPB

```
►►──┬────────────────────┬──►◄
     └─AVGPB=──bytes──────┘
```

For SDLC devices. Specifies the number of bytes used to control NCP buffer space for a polled device.

Code on: GROUP, LINE, PU (page 2-372).

BACKUP

```
      ┌─BACKUP=0───────┐
►►──┬─┴────────────────┴─┬──►◄
     └─BACKUP=count───────┘
```
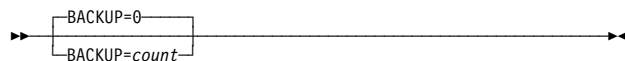
For XRF support. Specifies how many logical units will use XRF.

Code on: BUILD (page 2-11).

| **BALANCE**

```
►►──┬──────────────────────────────┬──►◄
     │               ┌─D─┐          │
     └─BALANCE=(factor,─┼─type─┼──)──┘
                     ├─D─┤
                     └─H─┘
```
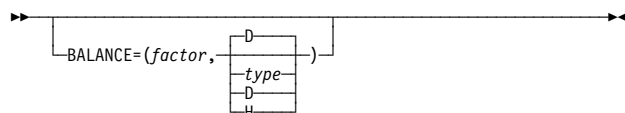
For 3745 frame-relay DLCIs. Specifies the balancing factor to be applied to the DLCI(s) indicated by the ID keyword.

Code on: DLCI (page 2-86).

BALANCE

```
      ┌─BALANCE=(0,D)───────────────┐
►►──┬─┴─────────────────────────────┴─┬──►◄
     │            ┌─0──────┐  ┌─D─┐   │
     └─BALANCE=(──┼────────┼,─┼─D─┼─)──┘
                  └─factor─┘  └─H─┘
```
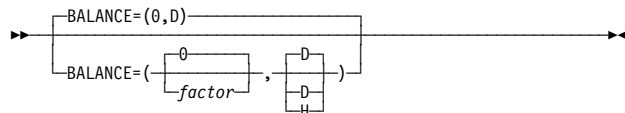
For NTRI physical lines. Specifies the connection balancing factor.

Code on: LINE (page 2-200).

BEACTO

```
      ┌─BEACTO=52───────┐
►►──┬─┴─────────────────┴─┬──►◄
     └─BEACTO=seconds──────┘
```
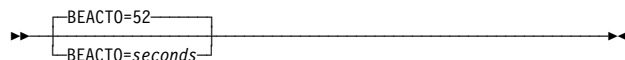
For Token Ring resources. Specifies how long NCP is to wait for a beaconing Token Ring to recover before deactivating all logical connections to it.

Code on: GROUP, LINE (page 2-201).

Keyword	Description

BFRDLAY

```
►►─┬──────────────────┬─►◄
   └─BFRDLAY=seconds──┘
```

For buffered terminals and nonswitched multipoint lines. Specifies the delay, in seconds, between successive transmissions to this device.

Code on: GROUP, LINE, CLUSTER, TERMINAL (page 2-452).

BFRS
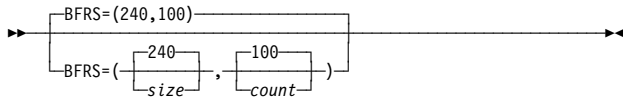
```
      ┌─BFRS=(240,100)───────────┐
►►────┤                          ├─►◄
      └─BFRS=(─┬─240──┬─,─┬─100──┬─)─┘
              └─size─┘   └─count─┘
```

Defines the size of the buffers in the NCP buffer pool and specifies the number of EP buffers to be allocated to the EP free buffer pool.

Code on: BUILD (page 2-11).

BHEXEC
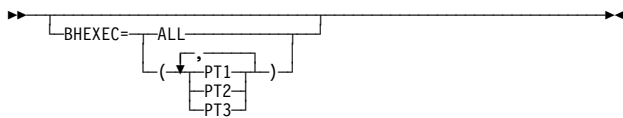
```
►►─┬──────────────────────┬─►◄
   │  ┌─ALL──────────┐    │
   └─BHEXEC=─┤              ├─┘
      └─(─◄┬─PT1─┬─)─┘
           ├─PT2─┤
           └─PT3─┘
```

For block handling routines. Specifies which block handling routines NCP is to use from the block handler set defined by the BHSET keyword.

Code on: GROUP, LINE, CLUSTER (page 2-67).

BHEXEC

```
      ┌─BHEXEC=PT1────┐
►►────┤               ├─►◄
      └─BHEXEC=─┬─PT1─┬─┘
               ├─PT2─┤
               └─PT3─┘
```
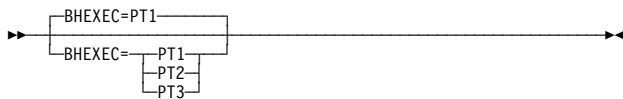
For block handling routines. Specifies the point at which this block handler is to run.

Code on: STARTBH (page 2-442).

BHEXEC
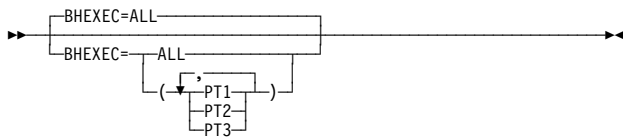
```
      ┌─BHEXEC=ALL───────────┐
►►────┤                      ├─►◄
      └─BHEXEC=─┬─ALL──────┬─┘
               └─(─◄┬─PT1─┬─)─┘
                    ├─PT2─┤
                    └─PT3─┘
```

For block handling routines. Specifies which block handling routines NCP is to use from the block handler set defined by the BHSET keyword.

Code on: TERMINAL (page 2-453), COMP.

BHSET

```
      ┌─BHSET=NONE──────┐
►►────┤                 ├─►◄
      └─BHSET=─┬─NONE────┬─┘
              ├─DYNAMIC─┤
              └─name────┘
```

For block handling routines. Associates a block handler set with a device.

Code on: GROUP, LINE, CLUSTER (page 2-68), TERMINAL (page 2-454), COMP.

BKSP

```
      ┌─BKSP=16────┐
►►────┤            ├─►◄
      └─BKSP=char──┘
```

Defines the text-canceling (backspace) character.

Code on: EDIT (page 2-87).

BLOCK
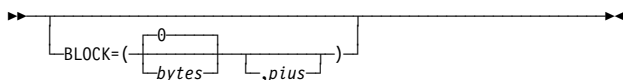
```
►►─┬────────────────────────────┬─►◄
   │         ┌─0─────┐           │
   └─BLOCK=(─┤       ├─┬───────┬─)─┘
            └─bytes─┘ └─,pius─┘
```

Specifies the maximum size, in bytes, for frames transmitted over this subarea link and the maximum number of PIUs for multiple-PIU frames.
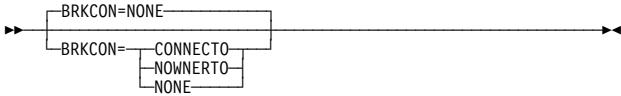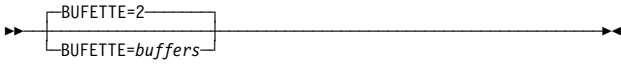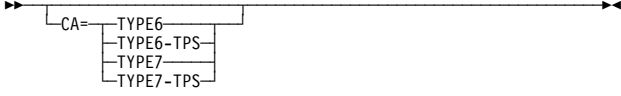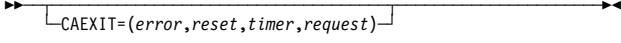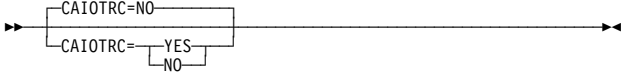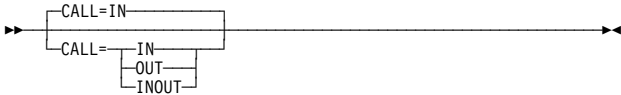
Code on: PU (page 2-373).

BNNSUP

```
►►─┬──────────────┬─►◄
   └─BNNSUP=3270──┘
```

For a type 1 physical unit on a nonswitched link. Specifies that this physical unit is an IBM 3270 operating in SDLC mode.

Code on: GROUP, LINE, PU (page 2-374).

Keyword	Description

BRANCH

```
    ┌─BRANCH=100─────┐
►►──┤                ├──►◄
    └─BRANCH=entries─┘
```

Specifies how many entries the branch trace table is to contain.

Code on: BUILD (page 2-12).

BRKCON

```
    ┌─BRKCON=NONE──────────┐
►►──┤                      ├──►◄
    └─BRKCON=──┬─CONNECTO─┬─┘
               ├─NOWNERTO─┤
               └─NONE─────┘
```

For switched subarea links. Specifies that NCP is to break a switched connection if a line has been idle for the time specified by the ACTIVTO keyword.

Code on: GROUP, LINE, PU (page 2-374).

BUFETTE

```
    ┌─BUFETTE=2───────┐
►►──┤                 ├──►◄
    └─BUFETTE=buffers─┘
```

For start-stop lines in emulation mode. Specifies how many 4-byte buffers are to be allocated to this line.

Code on: GROUP, LINE (page 2-201).

BUFSIZE

```
    ┌─BUFSIZE=64───┐
►►──┤              ├──►◄
    └─BUFSIZE=size─┘
```

For BSC lines in emulation mode. Specifies the size of the two cycle-steal buffers that are generated for each line in normal mode (CSPMODE=NORMAL).

Code on: GROUP, LINE (page 2-201).

CA

```
►►──┬──────────────────┬──►◄
    └─CA=──┬─TYPE6─────┬┘
           ├─TYPE6-TPS─┤
           ├─TYPE7─────┤
           └─TYPE7-TPS─┘
```

For channel links. Specifies the type of channel adapter installed in the communication controller in which NCP resides.

Code on: GROUP, LINE (page 2-202).

CAEXIT

```
►►──┬─────────────────────────────────────┬──►◄
    └─CAEXIT=(error,reset,timer,request)──┘
```

For channel user-written code. Specifies the entry points of user-written channel service routines.

Code on: GROUP (page 2-107).

CAIOTRC

```
    ┌─CAIOTRC=NO──────┐
►►──┤                 ├──►◄
    └─CAIOTRC=──┬─YES─┤
                └─NO──┘
```

Specifies whether NCP is to include the channel adapter I/O halfword trace.

Code on: BUILD (page 2-13).

CALINE

```
►►──┬──────────────┬──►◄
    └─CALINE=name──┘
```

For lines attached to an IBM modem with a data multiplexer. Specifies the name of the link attached to channel A of the modem.

Code on: GROUP, LINE (page 2-202).

CALL

```
    ┌─CALL=IN──────┐
►►──┤              ├──►◄
    └─CALL=──┬─IN────┤
             ├─OUT───┤
             └─INOUT─┘
```

For BSC, ISDN B-channel, or start-stop switched lines in network control mode. Specifies whether this line is to be used for incoming calls, outgoing calls, or both.

Code on: GROUP, LINE (page 2-203).

CHECK

Keyword	Description

CASDL

```
►►─┬─────────────────┬─►◄
   └─CASDL=seconds───┘
```

For channel links. Specifies how long the channel link can block inbound data before signaling that the station is inoperative.

Code on: GROUP, LINE (page 2-203), PU (page 2-374).

CATRACE

```
      ┌─CATRACE=(YES,50)──────────────────┐
►►────┤                                   ├─►◄
      │          ┌─YES─┐      ┌─50──────┐  │
      └─CATRACE=─┼(────┤      │         │─)┘
                 │     └─YES─┘└─entries─┘
                 └─NO─┘
```
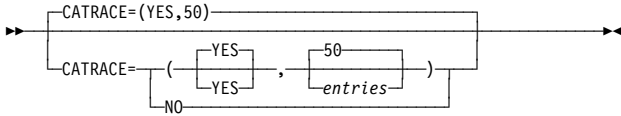
For the channel adapter trace facility. Specifies whether NCP is to include the channel adapter trace facility and how many entries the channel adapter trace table is to contain.

Code on: BUILD (page 2-13).

CCLID

```
      ┌─CCLID=NO──────┐
►►────┤               ├─►◄
      └─CCLID=─┬─YES─┬─┘
               └─NO──┘
```

For switched X.21 lines. Specifies whether NCP is to support called/calling line ID (CCLID).

Code on: GROUP, LINE (page 2-204).

CHANLA

```
►►─┬──────────────────┬─►◄
   └─CHANLA=─┬─YES─┬──┘
             └─NO──┘
```

For lines attached to an IBM modem with a data multiplexer. Specifies whether this line is attached to the first-level link LPDA control channel (channel A) on an IBM modem.

Code on: LINE (page 2-204).

CHANLNK

```
►►─┬───────────────────┬─►◄
   └─CHANLNK=─┬─YES─┬──┘
              └─NO──┘
```
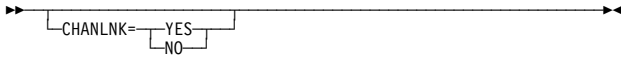
For user-written channel code. Specifies whether this group defines channel adapters for which user-written routines handle emulator subchannel operations.

Code on: GROUP (page 2-108).

CHANNELS

```
      ┌─CHANNELS=GROUP─┐
►►────┤                ├─►◄
      └─CHANNELS=GROUP─┘
```
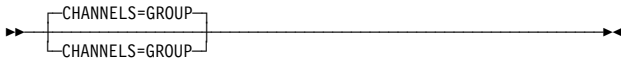
For the migration aid function. Specifies that 370 I/O channel attachments are to be defined on the GROUP, LINE and PU definition statements in the output generation definition (NEWDEFN file).

Code on: OPTIONS (page 2-342).

CHAREC

```
      ┌─CHAREC=XONOFF────────────────┐
►►────┤                              ├─►◄
      └─CHAREC=─┬─XONOFF───────────┬─┘
               │      ┌─XONOFF─┐   │
               └(─────┤        ├─,chars)┘
                      ├─XONOFF─┤
                      ├─XON────┤
                      ├─XOFF───┤
                      └─NO─────┘
```
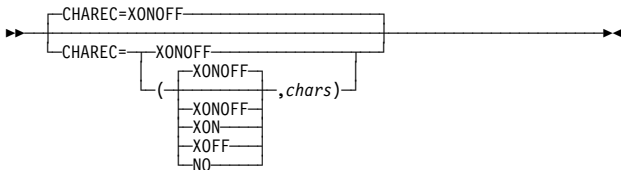
For TWX lines. Specifies the characters NCP or EP is to recognize as the end of a message when receiving data from a TWX terminal.

Code on: GROUP (page 2-108).

CHECK

```
      ┌─CHECK=NODCD────┐
►►────┤                ├─►◄
    │ └─CHECK=─┬─DCD───┬─┘
              └─NODCD─┘
```
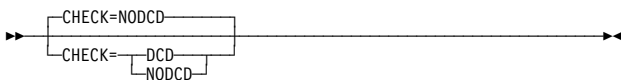
For switched, duplex, or start-stop lines in emulation mode. Specifies whether the control program is to use data-carrier-detect for this line.

Code on: LINE (page 2-205).

Keyword	Description

CHNLZ

```
     ┌─CHNLZ=NO──────────┐
►►───┤                   ├───►◄
     └─CHNLZ=─┬─YES─┬────┘
              └─NO──┘
```
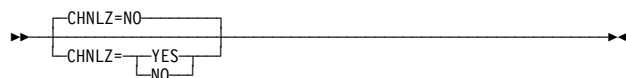
Specifies whether this line is attached to an IBM modem with a data multiplexer.

Code on: GROUP, LINE (page 2-205).

CHNPRI

```
     ┌─CHNPRI=NORMAL───────┐
►►───┤                     ├───►◄
     └─CHNPRI=─┬─NORMAL─┬──┘
               └─HIGH───┘
```
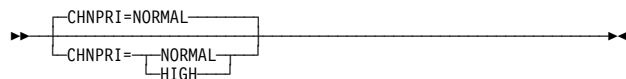
For BSC lines in emulation mode. Specifies the priority of the emulation subchannel associated with this line relative to the other emulation subchannels.

Code on: LINE (page 2-206).

| **CIR**

```
     ┌─CIR=(1,1,.1)────────────────┐
►►───┤                             ├───►◄
     └─CIR=(─┬─1──┬─,─┬─1──┬─,─┬─.1─┬─)─┘
            └─Bc─┘   └─Be─┘   └─Tc─┘
```
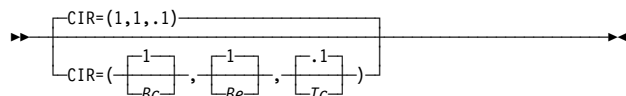
Specifies the CIR parameters for the DLCI(s) identified by the ID keyword of the DLCI statement.

Code on: DLCI (page 2-85).

| **CIR**
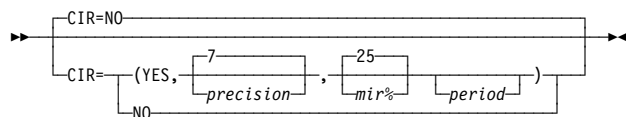
```
     ┌─CIR=NO─────────────────────────────────┐
►►───┤                                         ├───►◄
     └─CIR=─┬─(YES,─┬─7─────────┬─,─┬─25───┬─────────┬─)─┘
            │       └─precision─┘   └─mir%─┘└─period─┘ │
            └─NO────────────────────────────────────────┘
```

For 3746 Model 900 frame-relay and ISDN physical lines, specifies whether the Committed Information Rate (CIR) algorithms should be used for bandwidth management.

| **Code on:** GROUP, LINE (page 2-206).

CLINES
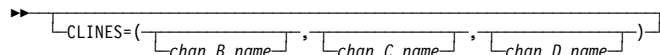
```
►►───┬─────────────────────────────────────────────────────────────┬───►◄
     └─CLINES=(─┬─────────────┬─,─┬─────────────┬─,─┬─────────────┬─)─┘
               └─chan_B_name─┘   └─chan_C_name─┘   └─chan_D_name─┘
```

For lines attached to an IBM modem with a data multiplexer. Specifies the names of the lines attached to channels B, C, and D of the IBM modem to which this line is attached.

Code on: LINE (page 2-207).

| **CLLM**

```
     ┌─CLLM=NO─────────────┐
►►───┤                     ├───►◄
     └─CLLM=─┬─(YES,─┬─11─┬─)─┐
            │       └─Ty─┘   │
            └─NO─────────────┘
```
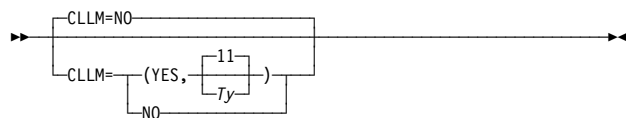
For 3746 Model 900 attached frame-relay physical lines. Specifies whether Consolidated Link Layer Management (CLLM) will be used.

| **Code on:** LINE (page 2-207).

CLOCKNG

```
►►───┬───────────────────────────┬───►◄
     └─CLOCKNG=─┬─INT──────────┬─┘
              ├─DIRECT───────┤
              └─(EXT─┬──────┬─)
                    └─,LIC─┘
```
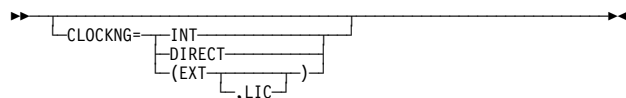
Specifies whether the modem or the communication scanner for the line provides clocking.

Code on: GROUP, LINE (page 2-208).

CLOCKNG

```
     ┌─CLOCKNG=INT─────────┐
►►───┤                     ├───►◄
     └─CLOCKNG=─┬─INT─┬────┘
               └─EXT─┘
```
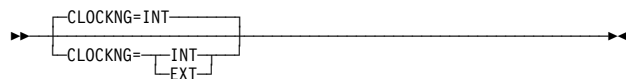
Specifies whether the communication scanner or the modem for this line provides clocking.

Code on: MTALCST (page 2-317).

COMPARE

Keyword	Description

CNLSQMAX

```
►►──┬─CNLSQMAX=10000──┬──────────────►◄
    └─CNLSQMAX=bytes──┘
```

For Internet Protocol. Specifies the maximum size for a session queue associated with an internet interface when the number of available buffers drops below the pseudo-slowdown threshold.

Code on: BUILD (page 2-13).

CNLSQTIM

```
►►──┬─CNLSQTIM=10────────┬─────────────►◄
    └─CNLSQTIM=seconds───┘
```
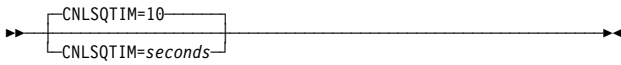
For Internet Protocol. Specifies how long NCP can keep a PIU on an outbound session queue associated with an internet interface.

Code on: BUILD (page 2-14).

CODE

```
►►──┬─CODE=──┬─ASCII─────┬──┬──────────►◄
            ├─BCD───────┤
            ├─BCD2──────┤
            ├─COR───────┤
            ├─COR2──────┤
            ├─DIC1──────┤
            ├─DIC3──────┤
            ├─EBCD──────┤
            ├─EBCDIC────┤
            ├─ITA2──────┤
            ├─KATAKANA──┤
            ├─USASCII───┤
            ├─USASCII-T─┤
            └─ZSC3──────┘
```
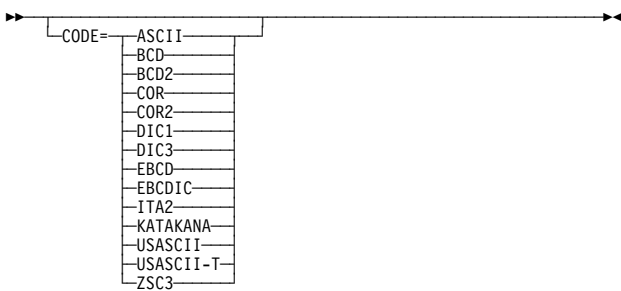
For start-stop and BSC lines. Specifies the transmission code to be used to communicate with devices attached to this line.

Code on: GROUP, LINE (page 2-210).

CODE

```
►►──┬─CODE=──┬─ASCII─┬──┬──────────────►◄
            ├─BCD───┤
            ├─BCD2──┤
            ├─COR───┤
            ├─COR2──┤
            ├─DIC1──┤
            ├─DIC3──┤
            └─EBCD──┘
```
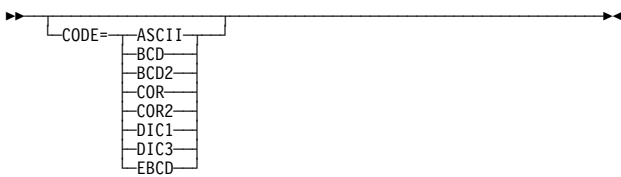
Specifies the transmission code to be used to communicate with the type of terminal defined by this MTALCST definition statement.

Code on: MTALCST (page 2-317).

CODE
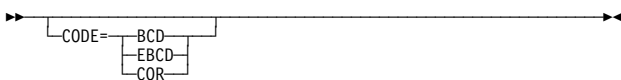
```
►►──┬─CODE=──┬─BCD──┬──┬───────────────►◄
            ├─EBCD─┤
            └─COR──┘
```

Specifies the transmission code used by the type of terminals represented in this MTA table.

Code on: MTATABL (page 2-326).

COMMAND
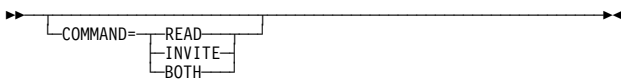
```
►►──┬─COMMAND=──┬─READ───┬──┬──────────►◄
              ├─INVITE─┤
              └─BOTH───┘
```

Specifies whether this user-written routine processes data received in response to a Read request, an Invite request, or both.

Code on: UBHR (page 2-475).

COMPACB

```
►►──┬─COMPACB=NO──────┬────────────────►◄
    └─COMPACB=──┬─YES─┬┘
              └─NO──┘
```

For IBM special products or user-written code. Specifies whether the UACBs for the lines in this group are compatible with the NCP ACBs for line trace, scanner interface trace, and wrap test.
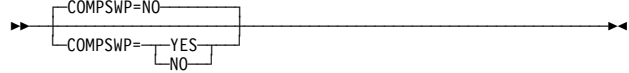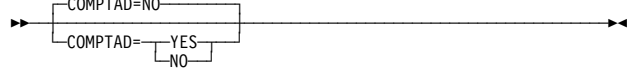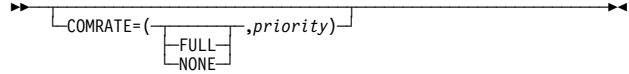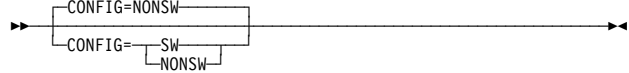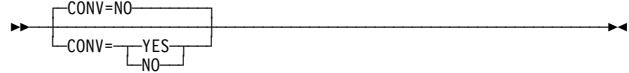
Code on: GROUP (page 2-109).

COMPARE

```
►►──┬─COMPARE=01─────┬─────────────────►◄
    └─COMPARE=chars──┘
```

For TWX terminals. Specifies, in hexadecimal, the compare character used during the line control selection table (LCST) entry selection process.

Code on: MTALCST (page 2-318).

Keyword	Description

COMPOWN

```
        ┌─COMPOWN=NO─────────┐
►►──────┤                    ├──────◄◄
        └─COMPOWN=─┬─YES─┬───┘
                   └─NO──┘
```

For IBM special products or user-written code. Specifies whether the control blocks for the lines in this group are compatible with the switchback, fallback, and LA disconnect functions.

Code on: GROUP (page 2-110).

COMPSWP

```
        ┌─COMPSWP=NO─────────┐
►►──────┤                    ├──────◄◄
        └─COMPSWP=─┬─YES─┬───┘
                   └─NO──┘
```

For IBM special products or user-written code. Specifies whether the control block structure for IBM special products or user-written code is compatible with the NCP control block structure for port swapping.

Code on: GROUP (page 2-111).

COMPTAD

```
        ┌─COMPTAD=NO─────────┐
►►──────┤                    ├──────◄◄
        └─COMPTAD=─┬─YES─┬───┘
                   └─NO──┘
```

For IBM special products or user-written code. Specifies whether the control block structure of IBM special products or user-written code is compatible with NCP for initialization, configuration data set update, bus switching initialization of TA and TD bus addresses, and initialization of the AIT index.

Code on: GROUP (page 2-112).

COMRATE

```
        ┌──────────────────────────────────┐
►►──────┤                                   ├──────◄◄
        └─COMRATE=(─┬──────┬─,priority)─────┘
                    ├─FULL─┤
                    └─NONE─┘
```

For frame-relay logical stations (FRTEs) and ISDN logical subarea and peripheral lines, specifies whether certain frames sent over this line can be discarded by the frame-relay network to expedite the processing of other frames when the network is congested. For frame-relay logical stations (FRTEs) and NTRI frame handler logical lines, also specifies the relative transmit priority for a TESP or an FHSP.

Code on: GROUP, LINE (page 2-211), PU (page 2-375).

CONFIG

```
        ┌─CONFIG=NONSW───────┐
►►──────┤                    ├──────◄◄
        └─CONFIG=─┬─SW────┬──┘
                  └─NONSW─┘
```

For subarea links. Specifies whether the SDLC subarea link between two communication controllers is switched or nonswitched.

Code on: GROUP, LINE (page 2-212).

CONV

```
        ┌─CONV=NO────────────┐
►►──────┤                    ├──────◄◄
        └─CONV=─┬─YES─┬──────┘
                └─NO──┘
```

Specifies whether NCP, upon receiving a message block from a device, is to send the device a message block in response instead of a positive acknowledgment.

Code on: GROUP, LINE, TERMINAL (page 2-454), COMP.

COPIES

```
        ┌─────────────────────┐
►►──────┤                     ├──────◄◄
        └─COPIES=count────────┘
```

For dynamic network identification. Identifies this network as a model network and specifies how many copies of this network are to be made available in the NVT.

Code on: NETWORK (page 2-332).

CSMHDRC

Keyword	Description

CORNUM

```
►►──┬─────────────────┬──►◄
     └─CORNUM=number──┘
```

For lines attached to an IBM modem with a data multiplexer. Specifies a 2-byte correlation number for this line.

Code on: GROUP, LINE (page 2-213).

| **CRATEDLY**

```
| ►►──┬──────────────┬──►◄
|      └─CRATEDLY=n──┘
```

| For congestion control on 3746 Model 900 frame-relay and ISDN physical lines.

| **Code on:** GROUP, LINE (page 2-214).

CRDLAY
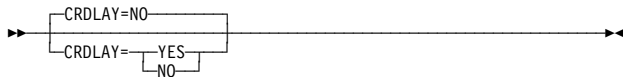
```
►►──┬─CRDLAY=NO──────────┬──►◄
     │           ┌─YES─┐  │
     └─CRDLAY=───┤     ├──┘
                 └─NO──┘
```

For start-stop lines. Specifies whether NCP is to delay processing a write operation after receiving a carriage return signal.

Code on: GROUP, LINE, TERMINAL (page 2-455).

CRETRY

```
►►──┬────────────────┬──►◄
     └─CRETRY=count──┘
```

For start-stop and BSC lines in network control mode. Specifies how many error recovery attempts NCP is to make when a recoverable error occurs on a line in this group.

Code on: GROUP (page 2-112).

CRITSIT

```
►►──┬─CRITSIT=NO──────────┬──►◄
     │            ┌─YES─┐  │
     └─CRITSIT=───┤     ├──┘
                  └─NO──┘
```

Specifies whether NCP is to notify a device, if it is logically active, when NCP is about to close down the network because of a critical situation.

Code on: GROUP, LINE, CLUSTER, TERMINAL (page 2-455).

CRRATE

```
►►──┬──────────────────────────┬──►◄
     └─CRRATE=print_positions──┘
```

For start-stop lines in network control mode. Specifies how many print positions the carriage returns for each idle character NCP sends to a printer on this line.

Code on: LINE (page 2-214).

CRRATE

```
►►──┬──────────────────────────┬──►◄
     └─CRRATE=print_positions──┘
```

Specifies how many print positions the carriage returns for each idle character NCP sends.

Code on: MTALCST (page 2-319).

CSMHDR

```
►►──┬─────────────────┬──►◄
     └─CSMHDR=header──┘
```

For BSC 3270 devices. Defines the header of the critical situation message to be sent to any IBM 3270 terminals in the network controlled by this NCP.
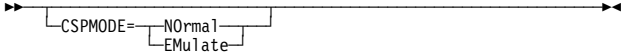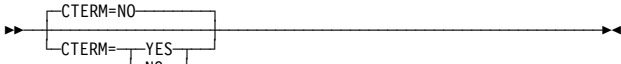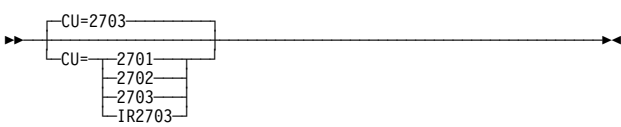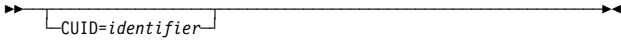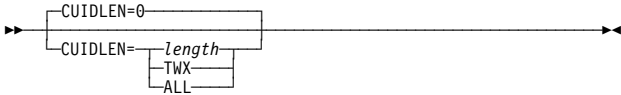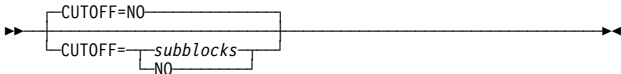
Code on: BUILD (page 2-14).

CSMHDRC

```
►►──┬───────────────────┬──►◄
     └─CSMHDRC=header──┘
```

For BSC 3270 terminals. Defines up to 111 additional characters for the critical situation message header defined by the CSMHDR keyword.

Code on: BUILD (page 2-14).

Keyword	Description

CSMSG

~~~
►►─┬─────────────────┬─►◄
   └─CSMSG=*message*─┘
~~~

For start-stop and BSC devices. Defines the text of the critical situation message to be sent to active stations on lines in network control mode before automatic network shutdown occurs.

Code on: BUILD (page 2-15).

CSMSGC

~~~
►►─┬──────────────────┬─►◄
   └─CSMSGC=*message*─┘
~~~

For start-stop and BSC devices. Defines up to 111 additional characters for the critical situation message defined by the CSMSG keyword.

Code on: BUILD (page 2-15).

CSPMODE

~~~
►►─┬────────────────────────┬─►◄
   └─CSPMODE=─┬─NOrmal──┬─┘
             └─EMulate─┘
~~~

For start-stop and BSC lines in emulation mode. Specifies the mode of operation for emulation mode lines.

Code on: GROUP, LINE (page 2-215).

CSSTIER

~~~
►►─┬──────────────────┬─►◄
   └─CSSTIER=─┬─A─┬─┘
             └─B─┘
~~~

For the 3746 Model 900 connectivity subsystem. Specifies the usage tier level for the optional NCP 3746 Model 900 Feature.

Code on: BUILD (page 2-15).

CTERM

~~~
   ┌─CTERM=NO──────┐
►►─┼───────────────┼─►◄
   └─CTERM=─┬─YES─┬─┘
           └─NO──┘
~~~

For switched lines. Specifies whether this TERMINAL definition statement defines devices that call the communication controller over the line associated with this definition statement.

Code on: TERMINAL (page 2-456).

CU

~~~
   ┌─CU=2703──────────┐
►►─┼──────────────────┼─►◄
   └─CU=─┬─2701───┬─┘
        ├─2702───┤
        ├─2703───┤
        └─IR2703─┘
~~~

For emulation mode. Specifies which transmission control unit functions NCP is to emulate for this line.

Code on: LINE (page 2-215).

CUID

~~~
►►─┬────────────────────┬─►◄
   └─CUID=*identifier*─┘
~~~

For switched BSC and TWX devices. Specifies the identification characters NCP is to send to BSC stations on lines operating in network control mode.

Code on: BUILD (page 2-16).

CUIDLEN

~~~
   ┌─CUIDLEN=0──────────┐
►►─┼────────────────────┼─►◄
   └─CUIDLEN=─┬─*length*─┬─┘
             ├─TWX────┤
             └─ALL────┘
~~~

For switched BSC and TWX devices. Specifies the length of the identifier sequence NCP is to send to this device when the device initiates a connection.

Code on: GROUP, LINE, TERMINAL (page 2-457).

CUTOFF

~~~
   ┌─CUTOFF=NO──────────┐
►►─┼────────────────────┼─►◄
   └─CUTOFF=─┬─*subblocks*─┬─┘
             └─NO────────┘
~~~

For start-stop and BSC lines in network control mode. Specifies the maximum number of subblocks NCP is to accept from a device attached to this line.

Code on: GROUP, LINE (page 2-215).

DELAY

Keyword	Description

CUTYPE

```
          ┌─CUTYPE=3271─────┐
►►────────┤                 ├────────────►◄
          └─CUTYPE=──┬─3271─┤
                     └─3275─┘
```
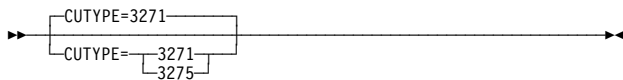
Specifies whether the control unit for this station is an IBM 3271 or IBM 3275.

Code on: GROUP, LINE, CLUSTER (page 2-68).

CUTYPE

```
          ┌─CUTYPE=3271─────┐
►►────────┤                 ├────────────►◄
          └─CUTYPE=──┬─3271─┤
                     ├─3275─┤
                     ├─2845─┤
                     └─2848─┘
```
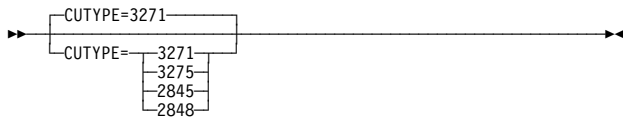
For start-stop and BSC lines in network control mode. Specifies the type of control unit to which the individual devices in a clustered station are attached.

Code on: GROUP, LINE (page 2-216).

CWALL

```
          ┌─CWALL=26──────┐
►►────────┤               ├──────────────►◄
          └─CWALL=buffers─┘
```

Specifies how many buffers NCP is to reserve to complete the flow of data through NCP.

Code on: BUILD (page 2-17).

DATABLK

```
          ┌─DATABLK=2048──┐
►►────────┤               ├──────────────►◄
          └─DATABLK=bytes─┘
```

For frame-relay, Token-Ring, and ISDN physical lines. Specifies the minimum guaranteed transmission bandwidth in bytes that can be allocated to a resource or to a protocol.

Code on: GROUP, LINE (page 2-217).

DATE

```
          ┌─DATE=YES─────┐
►►────────┤              ├───────────────►◄
          └─DATE=──┬─YES─┤
                   └─NO──┘
```
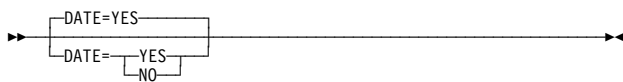
Specifies whether NCP is to insert the current date in a data block.

Code on: DATETIME (page 2-76).

DATEFMT

```
          ┌─DATEFMT=YY.DDD─────┐
►►────────┤                    ├──────────►◄
          └─DATEFMT=──┬─YY.DDD──┤
                      ├─MM/DD/YY─┤
                      ├─YY/MM/DD─┤
                      └─DD/MM/YY─┘
```
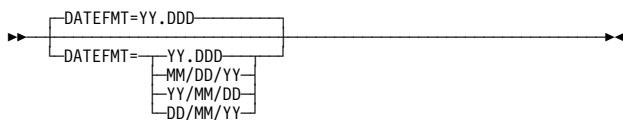
Specifies the format for the date inserted by the DATE keyword.

Code on: DATETIME (page 2-77).

DATMODE

```
►►──────┬─DATMODE=──┬─FULL─┬──────────────►◄
                    └─HALF─┘
```
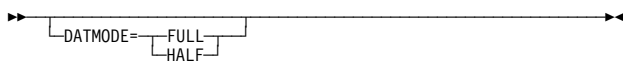
For nonswitched SDLC peripheral links. Specifies whether NCP is to communicate in half-duplex or duplex mode with this physical unit.

Code on: GROUP, LINE, PU (page 2-375).

DATRATE

```
          ┌─DATRATE=LOW──┐
►►────────┤              ├───────────────►◄
          └─DATRATE=──┬─LOW──┤
                      └─HIGH─┘
```
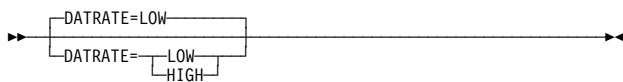
Specifies the data rate NCP is to use for a link attached to a dual-rate modem.

Code on: GROUP, LINE (page 2-217), MTALCST (page 2-319).

DELAY

```
          ┌─DELAY=NO──────┐
►►────────┤               ├───────────────►◄
          └─DELAY=──┬─600──┤
                    ├─1200─┤
                    └─NO───┘
```
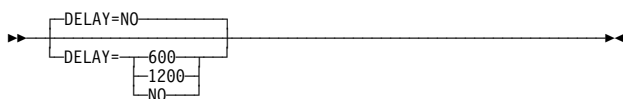
For WTTYs and start-stop lines in emulation mode. Specifies whether the teletypewriter terminals in this group require a line turnaround time of 70 to 80 milliseconds and, if so, their operation speed in bits per second.

Code on: GROUP (page 2-113).

Keyword	Description

DELAY

```
   ┌─DELAY=0.1─────┐
►►─┤               ├──►◄
   └─DELAY=seconds─┘
```

For channel links and ESCON logical link stations. Specifies how long NCP is to delay from the time data is available to send to the host to the time NCP presents an attention signal to the channel.

Code on: GROUP, LINE (page 2-217), PU (page 2-376).

DESTADDR

```
►►─┬──────────────────┬──►◄
   └─DESTADDR─┬─address─┬─┘
              └─0───────┘
```

For Internet Protocol. Specifies the internet address of the network, subnetwork, or host destination for an internet route.

Code on: IPROUTE (page 2-166).

DESTSA

```
           ┌─;─────────┐
►►─DESTSA=(─▼─address───┴─)──►◄
```

Specifies the subarea addresses of one or more destination subareas to which NCP routes PIUs using the explicit routes defined by this PATH definition statement.

Code on: PATH (page 2-355).

DIAL

```
   ┌─DIAL=NO──────┐
►►─┤              ├──►◄
   └─DIAL=─┬─YES─┬─┘
           └─NO──┘
```

Specifies whether the lines in this line group require switched line control procedures.

Code on: GROUP (page 2-113).

DIALALT

```
►►─┬───────────────┬──►◄
   └─DIALALT=name──┘
```

Specifies an alternate dial set for the dial set defined by this DIALSET definition statement.

Code on: DIALSET (page 2-80).

DIALALT

```
   ┌─DIALALT=NONE──────┐
►►─┤                   ├──►◄
   └─DIALALT=─┬─name─┬──┘
              └─NONE─┘
```

For nonswitched point-to-point and BSC multipoint lines. Specifies the name of the dial set NCP is to use to select an alternate line to communicate with a device.

Code on: GROUP, LINE (page 2-218).

DIALNO

```
►►─┬────────────────────────────┬──►◄
   └─DIALNO=(chars─┬─────────┬─)─┘
                   └─,length─┘
```

For switched lines. Specifies the dial digits and the length of the telephone number NCP is to use to call this device.

Code on: TERMINAL (page 2-458).

DIALSET

```
   ┌─DIALSET=NONE──────┐
►►─┤                   ├──►◄
   └─DIALSET=─┬─name─┬──┘
              └─NONE─┘
```

For call-out switched point-to-point BSC or start-stop lines. Specifies the name of the dial set for this line.

Code on: GROUP, LINE (page 2-218).

DIALSET

```
►►─┬────────────────────┬──►◄
   └─DIALSET=─┬─name─┬──┘
              └─NONE─┘
```

For call-out switched lines. Specifies the name of the dial set for this device.

Code on: TERMINAL (page 2-459).

Keyword	Description
DIALTO ►►─┬─DIALTO=60.0─┬─────────►◄ 　　　└─DIALTO=*seconds*─┘	For switched lines. Specifies how long NCP is to wait for an abandon-call-and-retry (ACR) signal from an automatic calling unit. **Code on:** BUILD (page 2-17).
DIRECTN ►►─┬──────────────────┬─►◄ 　　　└─DIRECTN=─┬─IN───┬─┘ 　　　　　　　　　　├─OUT──┤ 　　　　　　　　　　└─INOUT─┘	Specifies whether NCP will receive text from this station, send text, or both. **Code on:** GROUP, LINE, CLUSTER, TERMINAL (page 2-460).
DISABLE ►►─┬─DISABLE=NO──┬──────────►◄ 　　　└─DISABLE=─┬─YES─┬─┘ 　　　　　　　　　　└─NO──┘	For start-stop and BSC lines in emulation mode. Specifies whether the modem for this line requires a long disable interval when disconnecting from the line. **Code on:** GROUP, LINE (page 2-218).
DISP ►►─┬──────────────────┬─────►◄ 　　　└─DISP=─┬─NONPERM─┬─┘ 　　　　　　　　└─PERM────┘	Specifies whether the route being defined by this IPROUTE definition statement is dynamic (NONPERM) or permanent (PERM). **Code on:** IPROUTE (page 2-167).
DLCI ►►─┬────────────────────────────┬─►◄ 　　　└─DLCI=(──address──,─┬─D──┬─)─┘ 　　　　　　　　　　　　　　　├─H──┤ 　　　　　　　　　　　　　　　└─DD─┘	For frame-relay stations and NTRI frame handler logical links. Specifies the DLCI address of the frame-relay subport or internet host. All DLCI values associated with a single frame-relay physical line or a NTRI frame handler logical line must be unique. **Code on:** PU (page 2-377).
DPU ►►─┬─DPU=YES──┬──────────────►◄ 　　　└─DPU=─┬─YES─┬─┘ 　　　　　　　└─NO──┘	For the migration aid function. DPU specifies whether dynamic path update keywords are to be allowed to default, thereby adding dynamic path update capability to NCP. **Code on:** OPTIONS (page 2-343).
DR3270 ►►─┬──────────────────┬─────►◄ 　　　└─DR3270=─┬─YES─┬─┘ 　　　　　　　　　└─NO──┘	For dynamic reconfiguration. Specifies whether NCP is to support SDLC 3270 Model 11 and 12 terminals even though no such terminals are defined in your generation definition. This allows you to add these terminals through dynamic reconfiguration. **Code on:** BUILD (page 2-17).
DSABLTO ►►─┬─DSABLTO=3.0─┬──────────►◄ 　　　└─DSABLTO=*seconds*─┘	For network control mode. Specifies how long NCP is to wait for a modem to restore a failed data-set-ready signal. **Code on:** BUILD (page 2-18).
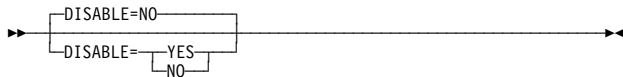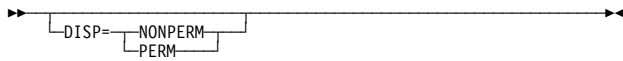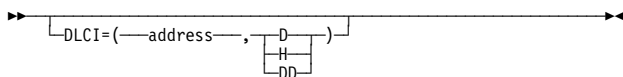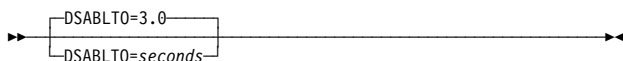
Keyword	Description

DSCRDDLY

```
►►──┬─────────────┬──────────────────────────►◄
    └─DSCRDDLY=n─┘
```

For congestion control on 3746 Model 900 ISDN and frame-relay physical lines. Specifies the DSCRDDLY threshold in seconds.

Code on: GROUP, LINE (page 2-219).

DSRTO

```
►►──┬─DSRTO=51──────┬──────────────────────────►◄
    └─DSRTO=seconds─┘
```

For leased BSC lines in emulation mode. Specifies how long NCP or EP is to wait for an emulator line to become data-set-ready before timing out on an ENABLE command.

Code on: GROUP (page 2-114).

DUALCOM
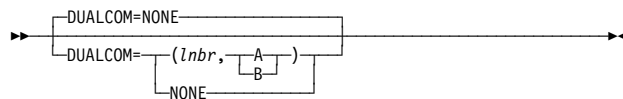
```
►►──┬─DUALCOM=NONE──────────────────┬──────────►◄
    └─DUALCOM=──┬─(lnbr,─┬─A─┬─)─┬──┘
                │        └─B─┘   │
                └─NONE───────────┘
```

For BSC lines in emulation mode. Specifies whether NCP or EP is to emulate the dual communication interface feature for this line when the line is operating in emulation mode.

Code on: LINE (page 2-219).

DUPADDR

```
►►──┬─DUPADDR=WARNING─┬──────────────────────────►◄
    └─DUPADDR=INFO────┘
```
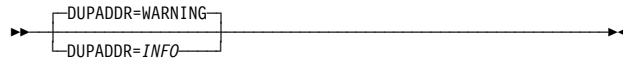
Specifies whether the ICN0261 message produced for duplicate poll addresses on the PU definition statement is to be an informational message or a warning message.

Code on: BUILD (page 2-18).

DUPLEX

```
►►──┬─DUPLEX=HALF───────┬──────────────────────────►◄
    └─DUPLEX=──┬─HALF─┬─┘
               └─FULL─┘
```
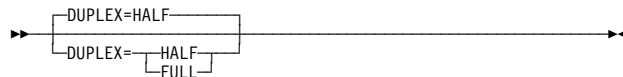
Specifies whether this communication line and modem constitute a half-duplex or a duplex facility.

Code on: GROUP, LINE (page 2-220).

DYNADMP

```
►►──┬─DYNADMP=NONE──────────┬──────────────────────────►◄
    └─DYNADMP=──┬─address─┬─┘
                └─NONE────┘
```
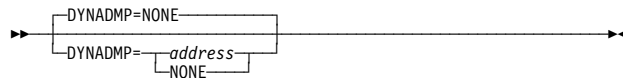
For channel links in emulation mode. Specifies whether the control program is to include the dynamic dump facility and, if so, its emulation subchannel address.

Code on: GROUP, LINE (page 2-221).

DYNNSC

```
►►──┬───────────────┬──────────────────────────►◄
    └─DYNNSC=count─┘
```

Specifies the maximum number of NPM session counter control blocks (NSCs) NCP can create dynamically.

Code on: BUILD (page 2-18).

DYNNSX

```
►►──┬───────────────┬──────────────────────────►◄
    └─DYNNSX=count─┘
```

Specifies the maximum number of NPM session counter extensions blocks (NSXs) NCP can create dynamically.

Code on: BUILD (page 2-18).

ENTRY

Keyword	Description

DYNPOOL
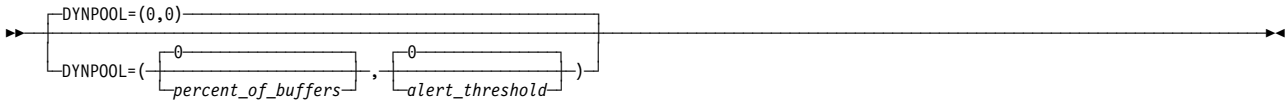
```
        ┌─DYNPOOL=(0,0)──────────────────────────────┐
►►──────┤                                            ├────────────────────►◄
        │           ┌─0────────────┐   ┌─0──────────┐│
        └─DYNPOOL=(─┤              ├─,─┤            ├─)┘
                    └─percent_of_buffers─┘   └─alert_threshold─┘
```

Defines the percent of NCP buffers available for dynamic allocation of control blocks and the associated alert threshold.

Code on: BUILD (page 2-19).

DYNWIND

```
        ┌─DYNWIND=(4,1,1,1,YES)──────────────────────┐
►►──────┤                                            ├────────────────────►◄
        │           ┌─4──┐   ┌─1──┐   ┌─1───┐   ┌─1──────┐   ┌─YES────┐
        └─DYNWIND=(─┤    ├─,─┤    ├─,─┤     ├─,─┤        ├─,─┤        ├─)┘
                    └─nw─┘   └─dw─┘   └─dwc─┘   └─dwdmax─┘   └─twcuse─┘
```
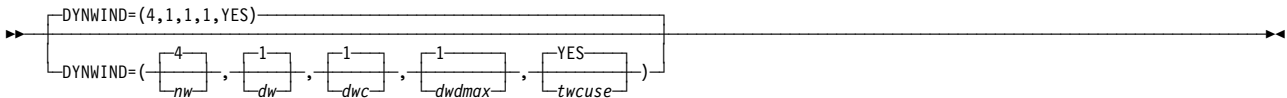
For Token-Ring, frame-relay, and ISDN physical lines. Specifies the values to be used by NCP to modify the IEEE 802.2 LLC working window parameter.

Code on: GROUP LINE (page 2-222), PU (page 2-378).

ECLTYPE

```
►►─┬──────────────────────────────────────────┬─►◄
   │            ┌─PHYsical─┐   ┌──────────────┐ │
   └─ECLTYPE=(──┤          ├─,─┤─PERipheral───├─)┘
                └─LOGical──┘   │─SUBarea──────│
                               │─ANY──(1)─────│
                               └─FRELAY─(2)───┘
```
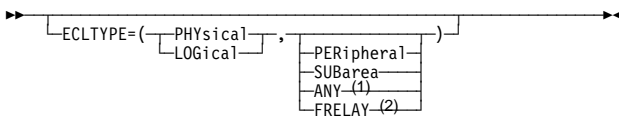
¹ Valid for physical lines only

² Valid for logical lines only

Required for Token-Ring resources. Specifies that the GROUP definition statement defines a physical or logical connection to a Token Ring.

Code on: GROUP (page 2-115).

ELEMENT

```
►►─┬──────────────────┬─►◄
   └─ELEMENT=address──┘
```

For predefined cross-network resource addresses. Defines the specific element address to be assigned to the resource if sequential addressing is not desired.

Code on: GWNAU (page 2-145).

ENABLTO

```
   ┌─ENABLTO=2.2──────┐
►►─┤                  ├─►◄
   └─ENABLTO=seconds──┘
```

For network control mode. Specifies how long NCP is to wait for a data-set-ready signal from a modem after the communication line attached to the modem is enabled or after a dialing operation is completed.

Code on: BUILD (page 2-20).

ENDTRNS

```
   ┌─ENDTRNS=EOT──────┐
►►─┤                  ├─►◄
   └─ENDTRNS=┬─EOT─┬──┘
             └─EOB─┘
```
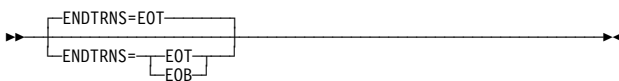
For start-stop lines. Specifies whether NCP is to recognize an end-of-transmission character (EOT) or an end-of-block character (EOB) as the end of a transmission.

Code on: GROUP, LINE, TERMINAL (page 2-460), COMP.

ENTRY

```
►►─┬──────────────┬─►◄
   └─ENTRY=name───┘
```

Specifies the entry point within the user-written module identified by the NAME keyword.

Code on: UBHR (page 2-475).

Keyword	Description

EOB

For WTTY lines in emulation mode. Specifies the character sequence that NCP or EP is to recognize as the EOB or end-of-message sequence when received from any WTTY terminal or certain U.S. and Canadian teletypewriter terminals.

Code on: GROUP (page 2-116).

EOT

For WTTY lines in emulation mode. Specifies the character sequence that NCP or EP is to recognize as the EOT sequence when received from any WTTY terminal or certain U.S. and Canadian teletypewriter terminals.

Code on: GROUP (page 2-116).

ERLIMIT
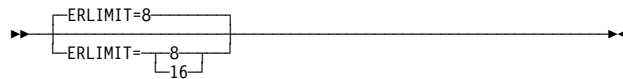
Specifies whether NCP is to support 8 or 16 explicit routes.

Code on: BUILD (page 2-20).

ERLIMIT

Specifies whether NCP is to support 8 or 16 explicit routes for this non-native network.

Code on: NETWORK (page 2-333).

ERn
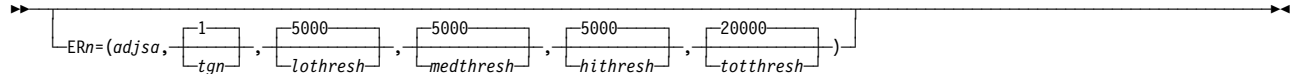
Specifies the adjacent subarea, transmission group number, and transmission group flow-control thresholds for explicit routes 0 to 15.

Code on: PATH (page 2-355).

ERRORT
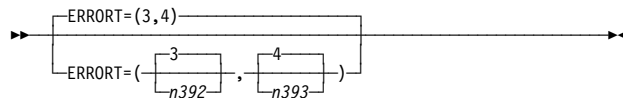
For frame relay and ISDN. Specifies the LMI error threshold at which NCP determines that communication with an adjacent frame-relay device has been lost.

Code on: GROUP, LINE, PU (page 2-380).

ETHERNET

Required for physical connections for Ethernet-type LANs. Specifies that the lines in this group are connected to an Ethernet-type LAN.

Code on: GROUP (page 2-117).

ETRATIO
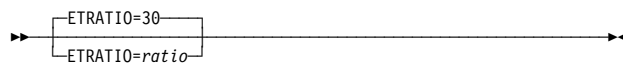
Specifies the error-to-transmission ratio to be used by the programs that analyze statistical maintenance records.

Code on: GROUP, LINE (page 2-224).

FEATURE

Keyword	Description

EXEC

```
        ┌─EXEC=YES─────────┐
►►──────┤                  ├──────►◄
        └─EXEC=─┬─YES─┬────┘
                └─NO──┘
```
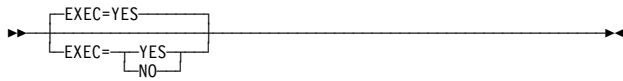
Specifies whether this block handler set can be run when it is associated with a device by command from the host processor.

Code on: BHSET (page 2-5)

EXEC

```
        ┌─EXEC=YES─────────┐
►►──────┤                  ├──────►◄
        └─EXEC=─┬─YES─┬────┘
                └─NO──┘
```
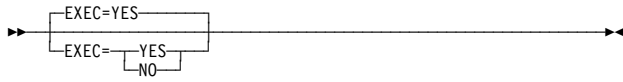
Specifies how the block handler is to be activated.

Code on: GROUP, LINE, CLUSTER (page 2-68), TERMINAL (page 2-460), COMP.

FANOUT

```
►►──┬────────────────┬──────►◄
    └─FANOUT=name─────┘
```

For BSC terminals. Specifies that this TERMINAL definition statement is the first in a sequence of TERMINAL definition statements representing devices attached to the same modem, and specifies the name of the last TERMINAL definition statement in the sequence.

Code on: TERMINAL (page 2-461).

FASTRUN

```
►►──┬──────────────┬──────►◄
    └─FASTRUN=ON────┘
```
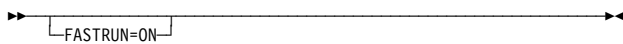
Specifies whether the NDF generation process is to generate table and link-edit source code.

Code on: OPTIONS (page 2-343).

FEATURE

```
        ┌─FEATURE=NOGPLKUP──────┐
►►──────┤                       ├──────►◄
        └─FEATURE=─┬─NOGPLKUP─┬─┘
                   └─GPLKUP───┘
```
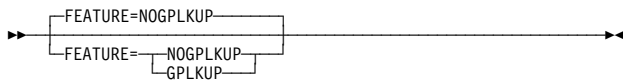
Specifies how NCP is to determine which device responds to a poll.

Code on: CLUSTER (page 2-69).

FEATURE

```
►►──┬─FEATURE=(⁽¹⁾─┬─NODUALCD─┬─,─┬─NOIMEND─┬─,─┬─LRC───┬─,─┬─NOSPACE─┬─,─┬─NSTXBCC─┬─)─┬──────►◄
    │              ├─DUALCODE─┤   ├─IMEND───┤   ├─LRC───┤   ├─SPACE───┤   ├─STXBCC──┤   │
    │              └─NODUALCD─┘   └─NOIMEND─┘   └─NOLRC─┘   └─NOSPACE─┘   └─NSTXBCC─┘   │
    └────────────────────────────────────────────────────────────────────────────────┘
```
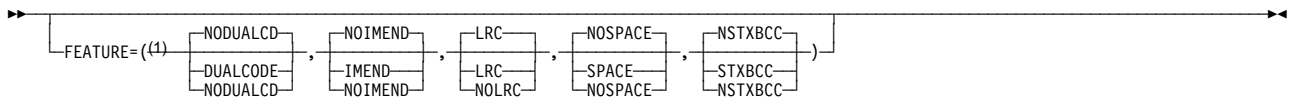
¹ Order does not matter.

For BSC and start-stop terminals. Specifies which terminal features NCP is to support.

Code on: GROUP, LINE (page 2-224).

FEATURE

```
►►──┬─FEATURE=(⁽¹⁾─┬─NOACR─┬─,─┬─NOATTN─┬─,─┬─NOBREAK─┬─,─┬─NOCHECK─┬─,─┬─NOSCTL─┬─,─┬─NOTOSUP──┬─,─┬─NOXCTL─┬─)─┬──────►◄
    │              ├─ACR───┤   ├─ATTN───┤   ├─BREAK───┤   ├─CHECK───┤   ├─SCTL───┤   ├─TOSUPPR──┤   ├─XCTL───┤   │
    │              └─NOACR─┘   └─NOATTN─┘   └─NOBREAK─┘   └─NOCHECK─┘   └─NOSCTL─┘   └─NOTOSUP──┘   └─NOXCTL─┘   │
    └───────────────────────────────────────────────────────────────────────────────────────────────────────┘
```
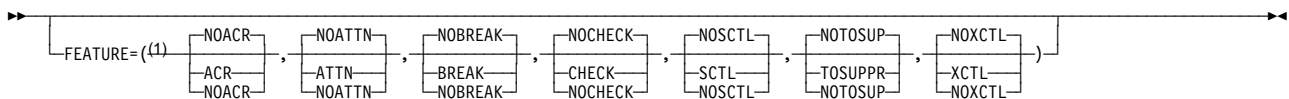
¹ Order does not matter.

For start-stop lines in network control mode. Specifies which terminal features NCP is to support.

Code on: TERMINAL (page 2-461).

Keyword	Description

FRAMECNT

```
        ┌─FRAMECNT=(100000,5000)──────────┐
►►──────┤                                 ├──►◄
        │          ┌─100000─┐  ┌─5000─┐   │
        └─FRAMECNT=(┤        ├,─┤      ├─) ┘
                    └─m──────┘  └─n────┘
```
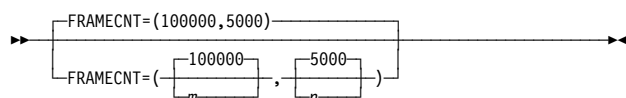
For Ethernet-type LANs. Specifies threshold values for the total number of frames transmitted and received and for the total number of frames that can be lost because of transmission and receive errors during an error cycle. When either threshold is reached, NCP generates an alert.

Code on: LINE (page 2-226).

FRAMING

```
        ┌─FRAMING=11─┐
►►──────┤            ├────────────────────►◄
        │          ┌─10─┐
        └─FRAMING=─┤    ├┘
                   └─11─┘
```

For TWX lines in network control mode. Specifies how many stop bits NCP is to send for each character and expects to receive from TWX terminals in this line group.

Code on: GROUP (page 2-117).

FRELAY

```
           ┌─PHYsical─┐  ┌───────────┐
►►─────────┤          ├,─┤─PERipheral─├─)──►◄
   └─FRELAY=(─LOGical─┘  │─SUBarea────│
                        └─ANY─(1)────┘
```

¹ Valid for physical lines only
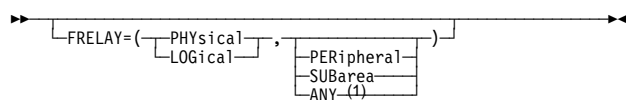
Required for frame relay. Specifies the type of frame-relay line.

Code on: GROUP (page 2-118).

FROM

```
►►──FROM=name──────────────────────────────►◄
```

Specifies the name of the LINE definition statement from which a physical unit is to be deleted, or the name of the physical unit from which a logical unit is to be deleted.

Code on: DELETE (page 2-79).

FRRATES

```
        ┌─FRRATES=(1,1,1)──────────────────────────────────┐
►►──────┤                                                  ├──►◄
        │        ┌─1──────────┐ ┌─1─────────┐ ┌─1─────────┐
        └─FRRATES=(┤          ├,┤           ├,┤           ├)┘
                  └─sna_priority┘ └─ip_priority┘ └─fh_priority┘
```
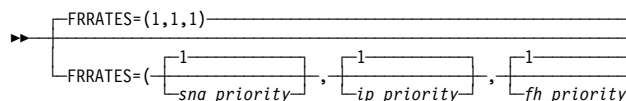
For 3745 frame relay physical lines. Specifies the relative transmission priority for SNA, IP, and frame-handler protocol traffic when supported by the physical line.

Code on: GROUP, LINE (page 2-227).

FRSEDRPU

```
►►─────────┬──────────────┬─────────────────►◄
           │  ┌─count──┐   │
           └─FRSEDRPU=─┤    ├┘
                       └─frdrcnt─┘
```
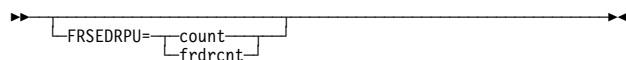
For frame-relay frame handler and NTRI frame handler logical subports. Specifies how many 3745 frame handler subports added to NCP through dynamic reconfiguration are to be eligible for NPM performance data collection.

Code on: PUDRPOOL (page 2-429).

FRSPRI

```
        ┌─FRSPRI=0─┐
►►──────┤          ├────────────────────────►◄
        └─FRSPRI=n─┘
```

For 3746 Model 900 congestion control. Specifies a relative service priority at a station level.

Code on: GROUP, LINE (page 2-227), PU (page 2-381).

GWPACING

Keyword	Description

GENILU

```
         ┌─GENILU=YES─┐
►►───────┤            ├──────────────────────►◄
         └─GENILU=─┬─YES─┬─┘
                   └─NO──┘
```

Specifies whether NDF is to generate control blocks and element addresses for independent logical units.

Code on: BUILD (page 2-21).

GENLEVEL

```
►►──┬──────────────────┬──────────────────────►◄
    └─GENLEVEL=string──┘
```

Specifies a character string that enables VTAM to load and activate NCP even if the date and time of NCP generation do not match the DTG control block and the type 1 RRT entry.

Code on: BUILD (page 2-21).

GPOLL

```
►►──┬──────────────┬──────────────────────────►◄
    └─GPOLL=chars──┘
```

Specifies that NCP is to use the general-polling procedure for this station and specifies the general-polling characters assigned to the control unit of the station.

Code on: CLUSTER (page 2-69).

GP3174

```
         ┌─GP3174=NO─────────────┐
►►───────┤                       ├──────────────►◄
         └─GP3174=─┬─group_address─┬─┘
                   └─NO────────────┘
```
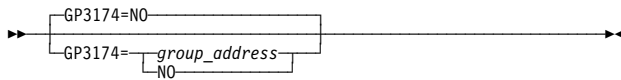
Specifies whether a poll to an IBM 3174 gateway controller is to be a group poll.

Code on: GROUP, LINE, PU (page 2-381).

GROUP

```
►►──GROUP=name───────────────────────────────►◄
```

Specifies the name of the stand-alone GROUP definition statement for a line group whose terminal characteristics are the same as those of the terminal that calls the communication controller over the MTA line.

Code on: MTALCST (page 2-320).

GROUP

```
►►──GROUP=name───────────────────────────────►◄
```

Specifies the name of the GROUP definition statement that contains the link parameters associated with this SDLC selection table entry.

Code on: SDLCST (page 2-434).

GWAEXIT

```
►►──┬──────────────┬──────────────────────────►◄
    └─GWAEXIT=name─┘
```

For a gateway NCP and user-written code. Specifies the entry point in the module that contains the user-written accounting exit routine for cross-network sessions.

Code on: BUILD (page 2-22).

GWPACING

```
         ┌─GWPACING=(0,ADAP,ALLOW)────────────────┐
►►───────┤                                        ├──►◄
         │              ┌─0────┐   ┌─ADAP──┐  ┌─ALLOW─┐  │
         └─GWPACING=(───┤      ├─,─┤       ├─,─┤       ├─)─┘
                        └─window─┘   ├─ADAP──┤  ├─ALLOW─┤
                                     └─FIXED─┘  └─FAIL──┘
```
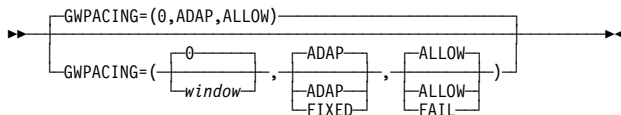
For a gateway NCP. Specifies the actions to take during a cross-network session if the BIND request or response indicates that session pacing will control the flow of data for the session.

Code on: BUILD (page 2-22), NETWORK (page 2-333).

Keyword	Description

GWSESAC

```
        ┌─GWSESAC=NO─────────────────────────────────────────────────────────────┐
►►──────┤                                                                         ├──►◄
        │            ┌─NODEFER─┐                      ┌──────┐  ┌─────────,──────┐ │
        └─GWSESAC=─┬─(YES,─┼─DEFER───┼─,─┬──────┬─,byteth,piuth,┬──────┬─,─▼─piudist─┬─,─┬─INITialpcid─┬─)─┘
                   │       └─NODEFER─┘   └─nscnum─┘              └─nscx─┘  └────────┘   └─────────────┘
                   └─NO──────────────────────────────────────────────────────────────┘
```
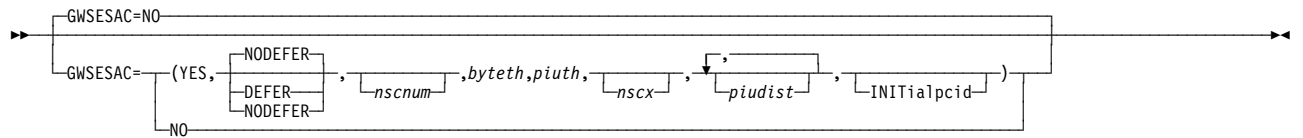
Specifies whether NCP is to include the gateway session accounting function and, if so, which session accounting characteristics are to be used.

Code on: BUILD (page 2-23).

HDWADDR

```
►►──HDWADDR=address──────────────────────────────►◄
```

Required for Internet Protocol over NTRI or Ethernet-type LANs. Specifies the hardware address of the internet host device defined by an IPHOST definition statement.

Code on: IPHOST (page 2-153).

HDXSP

```
        ┌─HDXSP=NO──────┐
►►──────┤               ├───────────────────────►◄
        └─HDXSP=─┬─YES─┬─┘
                 └─NO──┘
```
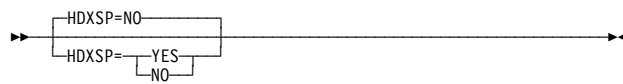
For SDLC lines. Specifies whether sending data to a terminal has priority over polling.

Code on: GROUP, LINE (page 2-228).

HICHAN

```
        ┌─HICHAN=NONE──────┐
►►──────┤                  ├────────────────────►◄
        └─HICHAN=─┬─address─┐
                  └─NONE────┘
```
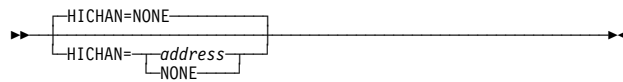
For channel links in emulation mode. Specifies the highest subchannel address on the channel adapter associated with any line operating in emulation mode or the address of the subchannel to be used for the dynamic dump facility.

Code on: GROUP, LINE (page 2-228).

HISPEED

```
►►─────┬─HISPEED=─┬─YES─┬─┬─────────────────────►◄
                  └─NO──┘
```
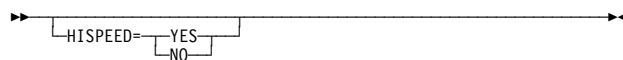
For BSC and TSS SDLC lines in network control mode. Specifies whether this line operates at speeds of 144 Kbps or greater.

Code on: GROUP, LINE (page 2-229).

| **HLOQSTOP**

```
| ►►─────┬─HLOQSTOP=n─┬──────────────────────────►◄
```

| For 3746 Model 900 frame-relay congestion control. Specifies the threshold that determines the maximum number of bytes that can be queued to a 3746 Model 900 attached frame relay physical line's hardware queue before a 3746 Model 900 software queue is created for the physical line.

| **Code on:** GROUP, LINE (page 2-229).

HOSTADDR

```
►►──HOSTADDR=host_address────────────────────────►◄
```

Specifies the internet address of the owning IBM TCP/IP host.

Code on: IPOWNER (page 2-161).

HSBPOOL

Keyword	Description

HOSTLINK

```
►►──┬────────────┬──►◄
    └─HOSTLINK=n─┘
```

For ESCON logical links. Assigns an identifier to a logical line so that it can be associated with a host link using the MOSS-E ESCA Link Information panel.

Code on: LINE (page 2-229).

HOSTRT

```
►►──┬─────────────────┬──►◄
    └─HOSTRT=─┬─NO──┬─┘
             └─YES─┘
```

Specifies whether this IPROUTE definition statement defines a route to a specific internet host or to an internet network or subnetwork.

Code on: IPROUTE (page 2-167).

HPR

```
     ┌─HPR=YES──────┐
►►──┼──────────────┼──►◄
    └─HPR=─┬─YES─┬─┘
          └─NO──┘
```
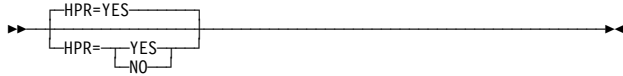
Specifies whether the HPR function is to be included in the NCP load module.

Code on: BUILD (page 2-25).

HPRATT

```
     ┌─HPRATT=12000─┐
►►──┼──────────────┼──►◄
    └─HPRATT=nnn───┘
```
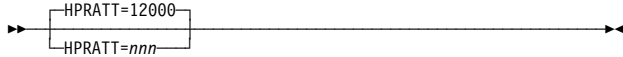
Specifies the average amount of time in microseconds that is required to transmit 1200 bits across a typical path in the composite ANR node's subarea network.

Code on: BUILD (page 2-26).

HPRMLC

```
     ┌─HPRMLC=9────┐
►►──┼─────────────┼──►◄
    └─HPRMLC=nnn──┘
```

Specifies the capacity in kilobits per second of the slowest subarea transmission group in the composite ANR node's subarea network that can carry APPN HPR data.

Code on: BUILD (page 2-27).

HPRMPS

```
     ┌─HPRMPS=0────┐
►►──┼─────────────┼──►◄
    └─HPRMPS=nnn──┘
```

Specifies the largest packet that can be sent across the composite ANR node's subarea network without being segmented on any of the subarea links along the path.

Code on: BUILD (page 2-28).

HPRQLIM
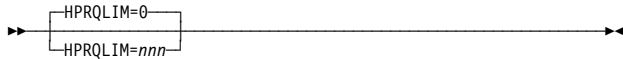
```
     ┌─HPRQLIM=0────┐
►►──┼──────────────┼──►◄
    └─HPRQLIM=nnn──┘
```

Specifies the maximum number of APPN HPR data bytes that can be queued to a peripheral link station's transmission queue at one time.

Code on: BUILD (page 2-30), GROUP, LINE, PU (page 2-382).

HPRSNAP

```
     ┌─HPRSNAP=NONE────────────────┐
►►──┼─────────────────────────────┼──►◄
    │         ┌─,──────┐           │
    └─HPRSNAP=─(─▼┬─SETUP─┬┬─)─────┘
    │           └─NLP───┘│
    └─────────────NONE───┘
```
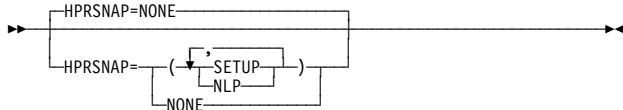
For APPN HPR. Specifies whether the snap trace function is to be enabled for HPR resources.

Code on: BUILD (page 2-30).

HSBPOOL

```
►►──┬────────────────┬──►◄
    └─HSBPOOL=hscbs──┘
```

For a gateway NCP. Specifies how many HSCBs the gateway NCP is to have for the SSCPs and logical units in any network that can participate in cross-network sessions.

Code on: BUILD (page 2-30).

Keyword	Description

ID

```
►►─┬──────────────────────────────┬─►◄
   │        ┌─D──┐                 │
   └─ID=─┬─(dlci,─┼─type─┼─)───┬───┘
         │        ├─D────┤     │
         │        ├─H────┤     │
         │        └─DD───┘     │
         ├─dlci₁–dlci₂──────────┤
         └─DEFAULT──────────────┘
```
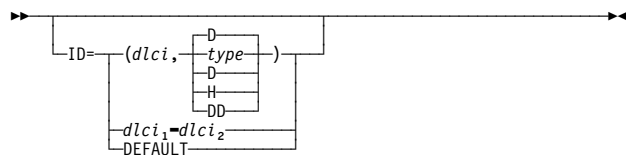
Identifies the DLCI to which the DLCI statement's keywords apply.

Code on: DLCI (page 2-84).

ID

```
►►─┬──────────────┬─►◄
   └─ID=nodeid────┘
```

For switched SDLC (LNCTL=SDLC) lines, ISDN B-channel lines, and switched frame-relay physical lines. Specifies the 4-byte hexadecimal node identification (NODEID) field to be returned to the host in a request contact request/response unit when a dial connection occurs.

Code on: GROUP, LINE (page 2-230).

IDNUM

```
►►─IDNUM=line_id────────────────►◄
```

Required when you code the SWMMT definition statement. Specifies a unique hexadecimal identifier for each line in a network for system user verification.

Code on: SWMMT (page 2-444).

IDSEQ

```
   ┌─IDSEQ=NONE───────┐
►►─┼──────────────────┼─►◄
   └─IDSEQ=─┬─name───┬─┘
            ├─NONE───┤
            ├─PASS───┤
            └─IGNORE─┘
```
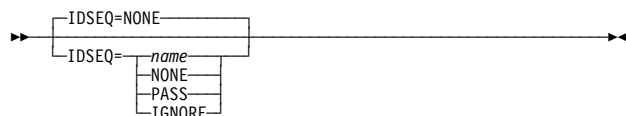
For the IBM 2740 Model 1 and for BSC and TWX devices on switched lines. Specifies whether this device transmits identification sequences when calling or being called by the communication controller and, if so, how NCP is to respond to the sequence.

Code on: GROUP, LINE, TERMINAL (page 2-463).

IDSEQ
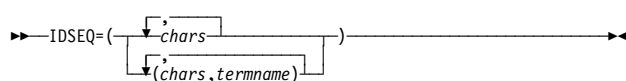
```
                  ┌─,──────┐
►►─IDSEQ=(─┬──▼─chars────────┬─)─►◄
          │   ┌─,─────────────┐  │
          └──▼─(chars,termname)──┘
```

Required when you code the IDLIST definition statement. Specifies the identification sequence for each device that can call or be called by the communication controller. Also specifies the name of the TERMINAL definition statement for the device the sequence is expected from.

Code on: IDLIST (page 2-149).

INBFRS

```
   ┌─INBFRS=3───────┐
►►─┼────────────────┼─►◄
   └─INBFRS=buffers─┘
```

For channel links. Specifies how many controller buffers NCP is to allocate for the first data transfer received from the host processor.

Code on: GROUP, LINE (page 2-230).

INCHI
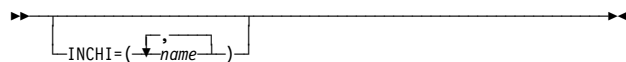
```
►►─┬────────────────────┬─►◄
   │        ┌─,───┐      │
   └─INCHI=(──▼─name──)──┘
```

Specifies the macro library members that contain linkage editor INCLUDE statements for object modules that are not level 2 or level 3 code. These modules run with storage protect key 1 and may reside anywhere in communication controller storage.

Code on: GENEND (page 2-96).

Keyword	Description

INCINIT
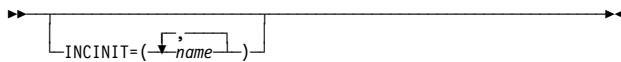
```
►►─┬──────────────────────────┬─►◄
   │        ┌─;─┐             │
   └─INCINIT=(─▼─name─┴─)─────┘
```

Specifies the macro library members that contain linkage editor INCLUDE statements for user-written initialization code.

Code on: GENEND (page 2-97).

INCLO

```
►►─┬──────────────────────┬─►◄
   │      ┌─;─┐           │
   └─INCLO=(─▼─name─┴─)───┘
```

Specifies the macro library members that contain linkage editor INCLUDE statements for object modules that are not level 2 or level 3 code. These modules run with storage protect key 0 and will reside in the low 64KB of communication controller storage.

Code on: GENEND (page 2-96).

INCL2HI
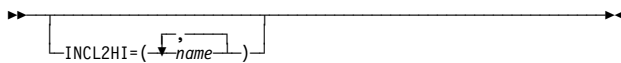
```
►►─┬─────────────────────────┬─►◄
   │       ┌─;─┐             │
   └─INCL2HI=(─▼─name─┴─)────┘
```

Specifies the macro library members that contain linkage editor INCLUDE statements for (1) object modules that are level 2 or level 3 code or (2) the user RECMS/NMVT module that NCP calls in level 5. These modules run with storage protect key 0 and may reside anywhere in communication controller storage.

Code on: GENEND (page 2-97).

INCL2LO
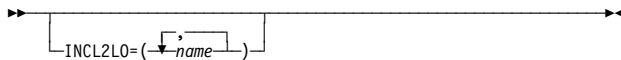
```
►►─┬─────────────────────────┬─►◄
   │       ┌─;─┐             │
   └─INCL2LO=(─▼─name─┴─)────┘
```

Specifies the macro library members that contain linkage editor INCLUDE statements for (1) object modules that are level 2 or level 3 code or (2) the user RECMS/NMVT module that NCP calls in level 5. These modules run with storage protect key 0 and will reside in the low 64KB of communication controller storage.

Code on: GENEND (page 2-97).

INHIBIT
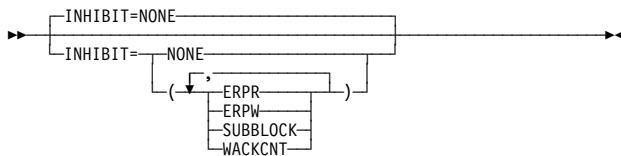
```
      ┌─INHIBIT=NONE────────────────────────┐
►►─┬──┴─────────────────────────────────────┴──┬─►◄
   └─INHIBIT=─┬─NONE──────────────────────┬─────┘
             │         ┌─,─┐             │
             └─(─▼─┬─ERPR─────┬─┴─)──────┘
                   ├─ERPW─────┤
                   ├─SUBBLOCK─┤
                   └─WACKCNT──┘
```

Specifies which NCP facilities are to be inhibited when NCP starts running after being loaded into the communication controller.

Code on: GROUP, LINE, CLUSTER (page 2-70).

INHIBIT
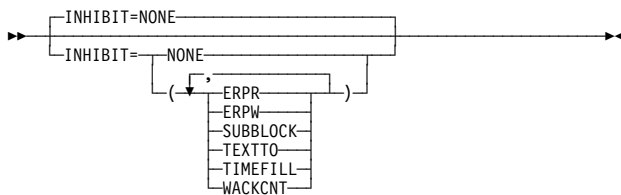
```
      ┌─INHIBIT=NONE────────────────────────┐
►►─┬──┴─────────────────────────────────────┴──┬─►◄
   └─INHIBIT=─┬─NONE──────────────────────┬─────┘
             │         ┌─,─┐             │
             └─(─▼─┬─ERPR─────┬─┴─)──────┘
                   ├─ERPW─────┤
                   ├─SUBBLOCK─┤
                   ├─TEXTTO───┤
                   ├─TIMEFILL─┤
                   └─WACKCNT──┘
```

Specifies which NCP facilities are to be inhibited when NCP starts running after being loaded into the communication controller.
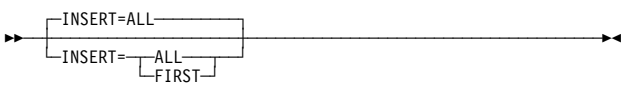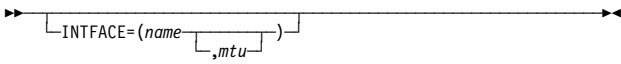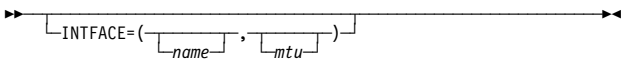
Code on: TERMINAL (page 2-464), COMP.

INIT

```
►►─┬───────────────────┬─►◄
   │    ┌─;─┐          │
   └─INIT=(─▼─name─┴─)─┘
```

Specifies the entry points of user-written, NTRI, or 3745 frame-relay initialization routines.

Code on: GENEND (page 2-94).

Keyword	Description

INNPORT

```
►►──┬────────────────────┬──►◄
     └─INNPORT=─┬─NO──┬───┘
                └─YES─┘
```

For frame relay, Token Ring, ISDN, and Ethernet-type LANs. Specifies whether a physical unit will remain active when CDLINK=ACT is specified on either the HALT or VARY INACT commands.

Code on: PU (page 2-382).

INSERT

```
     ┌─INSERT=ALL────────┐
►►──┼───────────────────┼──►◄
     └─INSERT=─┬─ALL───┬─┘
               └─FIRST─┘
```

Specifies whether NCP is to insert the date and time in just the first block of each message or in all blocks.

Code on: DATETIME (page 2-77).

INTFACE

```
►►──INTFACE=name──────────►◄
```

Specifies the name of the internet interface for which the local internet address is being defined.

Code on: IPLOCAL (page 2-156).

INTFACE

```
              ┌─,────┐
►►──INTFACE=(─▼─name─┴─)──►◄
```

Required for Internet Protocol. Specifies the name of the internet interface to be used to route IP datagrams to the owning TCP/IP host.

Code on: IPOWNER (page 2-161).

INTFACE

```
►►──INTFACE=name──────────►◄
```

For Internet Protocol. Specifies the name of the internet interface used to route IP datagrams to the network or subnetwork destination defined by DESTADDR on the same IPROUTE definition statement.

Code on: IPROUTE (page 2-167).

INTFACE

```
►►──┬─────────────────────────┬──►◄
     └─INTFACE=(name─┬──────┬─)─┘
                     └─,mtu─┘
```

For Internet Protocol. Specifies the name of the internet interface and the maximum transmission frame size for the line or logical unit being defined.

Code on: LINE (page 2-231), LU (page 2-299).

INTFACE

```
►►──┬────────────────────────────────┬──►◄
     └─INTFACE=(─┬──────┬─,─┬─────┬─)─┘
                 └─name─┘   └─mtu─┘
```

For Internet Protocol. Specifies the name of the internet interface and the maximum transmission frame size for the frame-relay interface, channel internet interface, or internal NCP to Model 900 point-to-point IP connection being defined.

Code on: PU (page 2-383).

IPADDR

```
►►──IPADDR=address─────────►◄
```

Required when you code the IPHOST definition statement. Specifies the internet address for the internet host defined by the IPHOST definition statement.

Code on: IPHOST (page 2-153).

IPL

```
     ┌─IPL=NO──────┐
►►──┼─────────────┼──►◄
     └─IPL=─┬─YES─┬─┘
            └─NO──┘
```

Specifies whether this NCP can be loaded and dumped over the link being defined.

Code on: GROUP, LINE (page 2-232).

ISDNCFG

Keyword	Description

IPPOOL

```
►►──┬────────────────────┬──►◄
    └─IPPOOL=─┬─NCP───┬──┘
             └─limit─┘
```

For Internet Protocol. Limits the number of buffers that can be used to receive IP datagrams.

Code on: BUILD (page 2-31).

IPQLIM

```
►►──┬──────────────┬──►◄
    └─IPQLIM=nnn──┘
```

For frame-relay internet resources. Specifies the maximum amount of data that can be queued on a frame-relay internet outbound queue awaiting transmission.

Code on: PU (page 2-384).

IPRATE

```
►►──┬────────────────────────┬──►◄
    └─IPRATE=(rate─┬───────┬─)─┘
                  └─,burst─┘
```
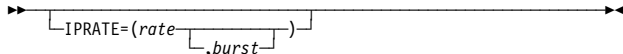
For Internet Protocol. Limits the rate at which NCP accepts internet frames.

Code on: BUILD (page 2-32).

IPSNAP
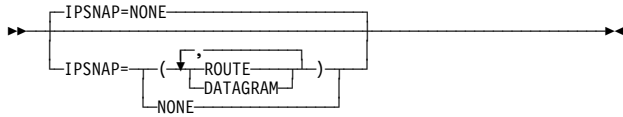
```
        ┌─IPSNAP=NONE───────────────┐
►►──────┤                           ├──►◄
        │           ┌─,──────┐      │
        └─IPSNAP=─┬─(─▼─ROUTE────┬─)─┘
                  │     └─DATAGRAM─┘
                  └─NONE──────────┘
```

For Internet Protocol. Specifies whether the snap trace function is to be enabled for internet resources.

Code on: BUILD (page 2-33).

IRETRY

```
        ┌─IRETRY=NO──────┐
►►──────┤                ├──►◄
        └─IRETRY=─┬─YES─┬─┘
                  └─NO──┘
```
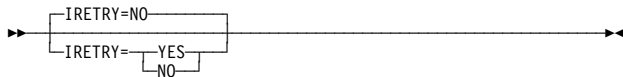
For nonswitched lines. Specifies whether NCP will repoll a secondary station before advancing to the next station in the SOT.

Code on: GROUP, LINE, PU (page 2-384).

ISDN
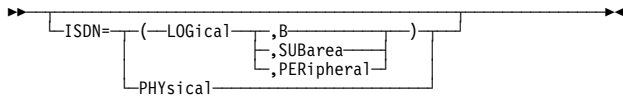
```
►►──┬──────────────────────────────────────┬──►◄
    └─ISDN=─┬─(─LOGical─┬──────────────┬─)─┘
            │          ├─,B──────────┤  │
            │          ├─,SUBarea────┤  │
            │          └─,PERipheral─┘  │
            └─PHYsical─────────────────┘
```

Required for ISDN resources. Specifies that the GROUP definition statement defines an ISDN physical or logical line group.

Code on: GROUP (page 2-119).

ISDNACC

```
        ┌─ISDNACC=YES────┐
►►──────┤                ├──►◄
        └─ISDNACC=─┬─YES─┬─┘
                   └─NO──┘
```
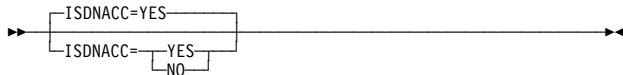
For ISDN resources. Specifies whether ISDN is to include ISDN accounting information.

Code on: BUILD (page 2-34).

ISDNCALL
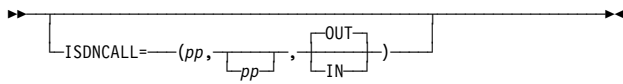
```
►►──┬───────────────────────────────────┬──►◄
    └─ISDNCALL=──(pp,─┬────┬─,─┬─OUT─┬─)─┘
                      └─pp─┘   ├─────┤
                               └─IN──┘
```

For 3746 Model 900 frame-relay physical lines. Specifies information for ISDN backup of the 3746 Model 900 frame-relay physical line.

Code on: LINE (page 2-232).

ISDNCFG

```
        ┌─ISDNCFG=CRC────┐
►►──────┤                ├──►◄
        └─ISDNCFG=─┬─CRC──┬─┘
                   └─NOCRC─┘
```

For ISDN physical lines. Specifies the ISDN channel configuration, specifically whether cyclic redundancy check (CRC) is included in frames to detect frame loss.

Code on: PU (page 2-385).

Keyword	Description

ISDNDSCT

```
►►──┬─────────────────────────┬──►◄
    └─ISDNDSCT=─┬─seconds─┬───┘
                └─NO──────┘
```

For ISDN physical lines. Specifies the time delay to bring down an ISDN B-channel after the last associated logical station disconnects.

Code on: PU (page 2-385).

ISDNLPN

```
►►──┬──────────────┬──►◄
    └─ISDNLPN=lpn──┘
```

For ISDN and 3746 Model 900 frame-relay physical lines. Specifies the local calling party number.

Code on: LINE (page 2-233).

ISDNLSA

```
►►──┬──────────────┬──►◄
    └─ISDNLSA=lsa──┘
```

For ISDN and 3746 Model 900 frame-relay physical lines. Specifies the local calling party subaddress.

Code on: PU (page 2-234).

ISDNRPN

```
►►──┬──────────────┬──►◄
    └─ISDNRPN=rpn──┘
```

For 3746 Model 900 frame-relay physical lines. Specifies the remote called party number of the ISDN backup line.

Code on: LINE (page 2-235).

ISDNRSA

```
►►──┬──────────────┬──►◄
    └─ISDNRSA=rsa──┘
```

For 3746 Model 900 frame-relay physical lines. Specifies the remote called party subaddress of the ISDN backup line.

Code on: LINE (page 2-235).

ITBMODE

```
►►──┬────────────────────────────────┬──►◄
    └─ITBMODE=(─┬─YES─┬──,──┬─YES─┬──)─┘
                └─NO──┘     └─NO──┘
```

For BSC devices in ITB mode. Specifies how NCP is to handle ITB characters in text transmitted between a station and the access method.

Code on: GROUP, LINE, CLUSTER (page 2-71), TERMINAL (page 2-465), COMP.

ITEXTTO

```
     ┌─ITEXTTO=NONE──────────┐
►►───┤                        ├──►◄
     └─ITEXTTO=─┬─seconds─┬──┘
                └─NONE────┘
```

For start-stop and BSC devices in network control mode. Specifies the text time-out interval NCP is to use for any terminal or component defined with INHIBIT=TEXTTO on the TERMINAL or COMP definition statement.

Code on: BUILD (page 2-34).

KBDLOCK

```
     ┌─KBDLOCK=80────┐
►►───┤                ├──►◄
     └─KBDLOCK=char──┘
```

For TWX lines in network control mode. Specifies, in hexadecimal, the control character NCP is to send to TWX terminals to signal operators to stop using the keyboard.

Code on: GROUP (page 2-120).

KEY0INC

```
►►──┬──────────────────────┬──►◄
    │            ┌─,──┐     │
    └─KEY0INC=(──▼─name─┴──)─┘
```

Specifies the macro library members that contain linkage editor INCLUDE statements for object modules to be included in read-only storage.

Code on: GENEND (page 2-98).

LCTYPE

Keyword	Description

KEY0ORD
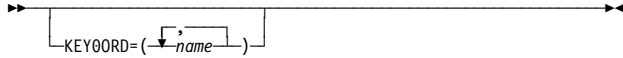
```
►►─┬──────────────────────┬─►◄
   └─KEY0ORD=(─┬─;─┬─name─┬─)─┘
              └───┘
```

Specifies the macro library members that contain linkage editor ORDER statements for object modules to be included in read-only storage.

Code on: GENEND (page 2-98).

LADDR

```
►►─┬──────────────────────────────┬─►◄
   └─LADDR=(──address──,─┬─D──┬─)─┘
                        ├─H──┤
                        └─DD─┘
```
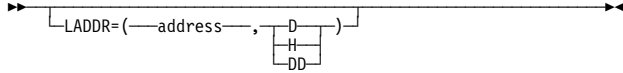
Required for Internet Protocol. Specifies the internet address of the associated gateway interface.

Code on: IPLOCAL (page 2-156).

LADDR
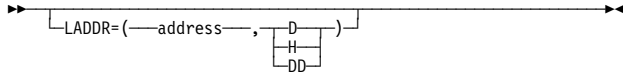
```
►►─┬──────────────────────────────┬─►◄
   └─LADDR=(──address──,─┬─D──┬─)─┘
                        ├─H──┤
                        └─DD─┘
```

Frame-relay and channel internet resources, and internal NCP to Model 900 point-to-point IP connections. Specifies the internet address of the associated interface.

Code on: PU (page 2-385).

LANTYPE

```
►►─┬──────────────────────────┬─►◄
   └─LANTYPE=─┬─ENETV2───┬─────┘
             ├─802.3───┤
             └─DYNAMIC─┘
```
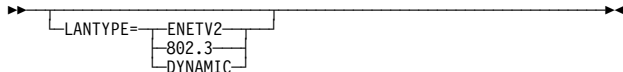
Required for Ethernet-type LANs. Specifies the type of frame format this physical unit supports.

Code on: GROUP, LINE, PU (page 2-386).

LANTYPE
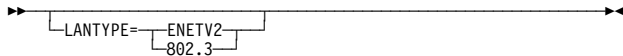
```
►►─┬────────────────────────┬─►◄
   └─LANTYPE=─┬─ENETV2─┬─────┘
             └─802.3──┘
```

Specifies the type of frame format used by the internet host attached to an Ethernet-type LAN.

Code on: IPHOST (page 2-154).

LCST

```
   ┌─LCST=NONE─────────┐
►►─┼───────────────────┼─►◄
   └─LCST=─┬─name─┬─────┘
           └─NONE─┘
```

For MTA lines. Specifies the name of the MTALCST definition statement that defines the device characteristics NCP uses when calling the device defined by this TERMINAL definition statement.

Code on: GROUP, LINE, TERMINAL (page 2-465).

LCST

```
►►─LCST=(─┬─;─┬─name─┬─)─────►◄
          └───┘
```

Specifies the name of up to ten MTALCST definition statements to be associated with this MTA table.

Code on: MTATABL (page 2-327).

LCTYPE

```
   ┌─LCTYPE=1050────────┐
►►─┼────────────────────┼─►◄
   └─LCTYPE=─┬─1050──┬───┘
             ├─2740A─┤
             ├─2740D─┤
             ├─2740E─┤
             ├─2740F─┤
             ├─2741──┤
             ├─3767──┤
             └─TWX───┘
```
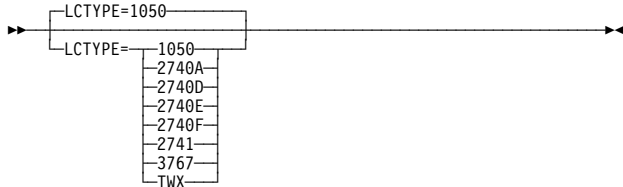
Specifies the type of terminal and line control to be used.
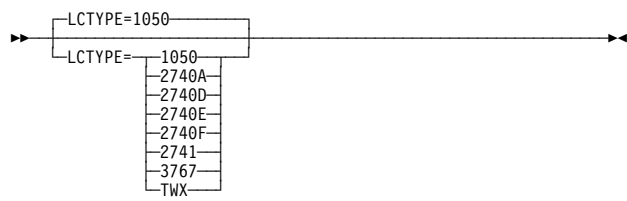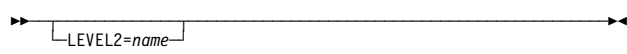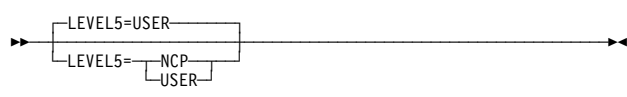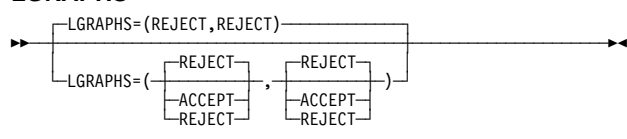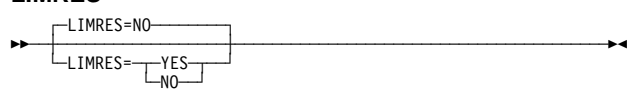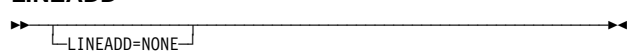
Code on: MTALCST (page 2-320).

LCTYPE

```
►►─LCTYPE=(─┬─;─┬─type─┬─)─────►◄
            └───┘
```

Specifies the terminal types included in this list of terminal identification procedures.

Code on: MTALIST (page 2-323).

Keyword	Description

LCTYPE

```
      ┌─LCTYPE=1050─────┐
►►────┤                 ├──────────────────────►◄
      └─LCTYPE=─┬─1050──┬─┘
                ├─2740A─┤
                ├─2740D─┤
                ├─2740E─┤
                ├─2740F─┤
                ├─2741──┤
                ├─3767──┤
                └─TWX───┘
```

Specifies the type of line control used by the terminals represented in this MTA table.

Code on: MTATABL (page 2-327).

LENAME

```
      ┌─LENAME=INLINKED─┐
►►────┤                 ├──────────────────────►◄
      └─LENAME=name─────┘
```

For VSE or DOS only. Specifies the name of the member containing the link-edit control statements.

Code on: BUILD (page 2-34), GROUP, LINE, TERMINAL.

LEVEL2

```
►►──┬────────────────┬──────────────────────────►◄
    └─LEVEL2=name────┘
```

For user-written line control, NTRI, and 3745 frame relay. Specifies the entry point of user-written, NTRI, or 3745 frame-relay code for level 2 scanner interrupts.

Code on: GROUP (page 2-120).

LEVEL3

```
►►──┬────────────────┬──────────────────────────►◄
    └─LEVEL3=name────┘
```

For user-written line control, NTRI, and 3745 frame relay. Specifies the entry point of user-written, NTRI, or 3745 frame-relay code for level 3 interrupts.

Code on: GROUP (page 2-121).

LEVEL5

```
      ┌─LEVEL5=USER─┐
►►────┤             ├──────────────────────────►◄
      └─LEVEL5=─┬─NCP──┬─┘
                └─USER─┘
```

For user-written line control, NTRI, and 3745 frame relay. Specifies whether level 5 code is to be provided by NCP or is user-written.

Code on: GROUP (page 2-122).

LGRAPHS

```
      ┌─LGRAPHS=(REJECT,REJECT)──────────┐
►►────┤                                  ├──────►◄
      └─LGRAPHS=(─┬─REJECT─┬─,─┬─REJECT─┬─)─┘
                  ├─ACCEPT─┤   ├─ACCEPT─┤
                  └─REJECT─┘   └─REJECT─┘
```

Specifies whether NCP is to accept or reject leading graphics received from a station.

Code on: GROUP, LINE, CLUSTER (page 2-71), TERMINAL (page 2-466), COMP.

LIMRES

```
      ┌─LIMRES=NO──┐
►►────┤            ├──────────────────────────►◄
      └─LIMRES=─┬─YES─┬─┘
                └─NO──┘
```

Specifies whether this line is for a limited resource session that should be deactivated automatically when session traffic ceases on a switched link.

Code on: GROUP, LINE (page 2-236).

LINEADD

```
►►──┬──────────────┬──────────────────────────►◄
    └─LINEADD=NONE─┘
```

For user-written line control, NTRI, and 3745 frame relay. Specifies that no physical address interface is defined for this line group.

Code on: GROUP (page 2-123).

LLERP

Keyword	Description

LINEAUT

```
►►─┬───────────────┬─►◄
   └─LINEAUT=YES───┘
```

For user-written line control, Token-Ring, frame-relay, and ISDN resources. Specifies automatic calling for a line group for which no physical address interface is defined.

Code on: GROUP (page 2-123).

LINECB

```
►►─┬────────────────┬─►◄
   └─LINECB=name────┘
```

For user-written code and Internet Protocol. Specifies the name of the user-defined control block associated with this virtual link.

Code on: GROUP, LINE (page 2-236).

LINEFVT

```
►►─┬─────────────────┬─►◄
   └─LINEFVT=name────┘
```

For user-written code and Internet Protocol. Specifies the name of the FVT associated with this virtual link.

Code on: GROUP, LINE (page 2-236).

LINES

```
             ┌─,──────┐
►►─LINES=(───▼─name──┴─)─►◄
```

Required when you code the DIALSET definition statement. Specifies the switched lines that compose the dial set.

Code on: DIALSET (page 2-81).

LINESIZ

```
►►─┬──────────────────┬─►◄
   └─LINESIZ=length───┘
```

For start-stop lines in network control mode. Specifies the length of the print line for printer devices.

Code on: GROUP, LINE (page 2-237), MTALCST (page 2-321).

LINETRC

```
   ┌─LINETRC=(YES,4,4)──────────────────────────┐
►►─┴─────────────────────────────────────────────┴──►◄
   └─LINETRC=─┬─(─┬─YES─┬─,─┬─4─────┬─,─┬─4──────┬─)─┬──
              │   └─YES─┘   └─sits──┘   └─blocks─┘   │
              └─NO─────────────────────────────────┘
```
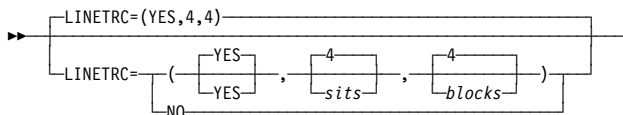
For BSC and start-stop devices in emulation mode. Specifies whether NCP is to include the line trace and scanner interface trace.

Code on: BUILD (page 2-35).

LINOPT

```
►►─┬───────────────┬─►◄
   └─LINEOPT=BSC───┘
```

Specifies that BSC line control modules are to be included in the NCP load module.
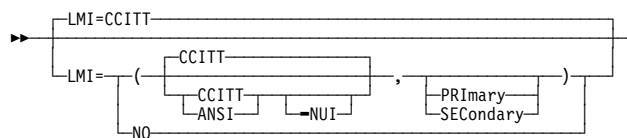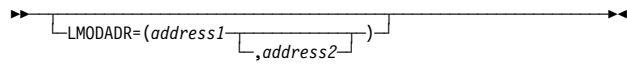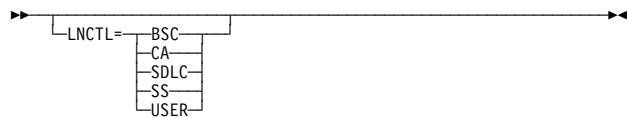
Code on: GENEND (page 2-98).

LLERP

```
►►─┬────────────────────┬─►◄
   └─LLERP=─┬─REQUIRED─┬─┘
            └─NOTPREF──┘
```

Specifies whether the link station will perform error recovery procedures for HPR data that flow on an RTP connection.

Code on: GROUP, LINE, PU (page 2-387).

Keyword	Description

LMI

```
         ┌─LMI=CCITT──────────────────────────────────┐
►►───────┤                                            ├──────────────────►◄
         │        ┌─CCITT────────┐   ┌─────────┐      │
         └─LMI=─┬─(─┼─CCITT─┬─┬──────┼─,─┬─PRImary───┬─)─┘
                │   └─ANSI──┘ └─-NUI─┘   └─SECondary─┘
                └─NO─────────────────────────────────┘
```

For frame-relay link station subports, ISDN physical PUs, and NTRI. Defines the LMI protocol standard and NCP echo detection for the link station subport.

Code on: GROUP, LINE, PU (page 2-388).

LMODADR

```
►►───┬──────────────────────────────────┬──────────────────►◄
     └─LMODADR=(address1─┬─────────────┬─)─┘
                         └─,address2───┘
```

Specifies the local modem addresses used on the line attached to the physical unit or terminal.

Code on: CLUSTER (page 2-72), PU (page 2-389), TERMINAL (page 2-466).

LNCTL

```
►►───┬──────────────────┬──────────────────►◄
     └─LNCTL=─┬─BSC──┬───┘
              ├─CA───┤
              ├─SDLC─┤
              ├─SS───┤
              └─USER─┘
```

Specifies the line control to be used for the lines in this group.

Code on: GROUP (page 2-124).

LNKOWNER

```
►►───┬──────────────────┬──────────────────►◄
     └─LNKOWNER=name─────┘
```

Specifies the name of the user-written generation load module that performs the link-level processing of the resources defined by this GROUP definition statement.

Code on: GROUP (page 2-125).

LNQTCNT

```
     ┌─LNQTCNT=1──────┐
►►───┼────────────────┼──────────────────►◄
     └─LNQTCNT=count──┘
```

For start-stop lines in network control mode. Specifies how many times NCP is to test a start-stop line after receiving data to ensure that the line is electrically quiet.

Code on: GROUP, LINE (page 2-237).

LOCADD

```
►►───┬──────────────────┬──────────────────►◄
     └─LOCADD=address────┘
```

Required for Token-Ring, frame-relay BAN connections, ISDN peripheral BAN connections, and Ethernet-type LAN physical lines. Specifies the locally administered address for a line connected to a LAN.

Code on: LINE (page 2-238).

LOCADDR

```
►►───LOCADDR=address─────────────────────────►◄
```

Specifies, in decimal, the local address of this logical unit.

Code on: LU (page 2-300).

LOCALTO

```
►►───┬──────────────────┬──────────────────►◄
     └─LOCALTO=seconds───┘
```

For Token-Ring logical lines. Specifies the Token-Ring reply timer (T1) value for stations connected to a local Token Ring.

Code on: BUILD (page 2-35).

Keyword	Description

LOCALTO

```
►►──┬─────────────────────┬──►◄
    └─LOCALTO=seconds─┘
```

For NTRI and frame-relay logical resources. Specifies the Token Ring reply timer (T1) value for stations connected to a frame-relay network or a local Token Ring.

Code on: GROUP, LINE (page 2-239), PU (page 2-389).

LOCHAN

```
      ┌─LOCHAN=NONE───────┐
►►──┼───────────────────┼──►◄
    └─LOCHAN=─┬─address─┬─┘
             └─NONE────┘
```
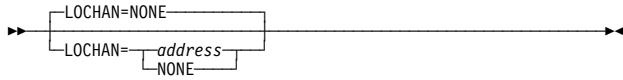
For channel links with emulation mode subchannels. Specifies the lowest subchannel address on the channel adapter associated with any line operating in emulation mode.

Code on: GROUP, LINE (page 2-240).

LPDA

```
►►──┬──────────────────┬──►◄
    └─LPDA=─┬─BLOCK─┬─┘
           └─ALLOW─┘
```

Specifies whether LPDA tests can be run for this physical unit.

Code on: PU (page 2-390).

LPDATS

```
      ┌─LPDATS=NO───────────────────┐
►►──┼─────────────────────────────┼──►◄
    └─LPDATS=─┬─LPDA2─────────────┬─┘
             ├─(LPDA1⁽¹⁾─┬──────┬─)─┤
             │          └─,3867─┘   │
             └─NO───────────────────┘
```
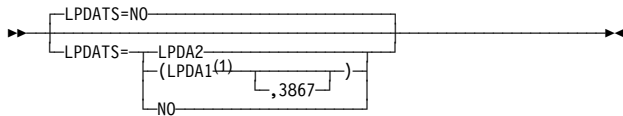
¹ Valid only for TSS- and HPTSS-attached lines

For BSC and SDLC lines. Specifies whether this line uses modems that support LPDA commands and whether these modems operate in normal or migration mode.

Code on: GROUP, LINE (page 2-240).

LPDA2DL

```
      ┌─LPDA2DL=NO──────┐
►►──┼─────────────────┼──►◄
    └─LPDA2DL=─┬─YES─┬─┘
             └─NO──┘
```
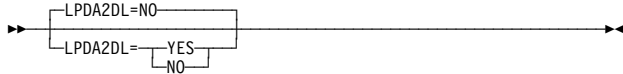
For BSC and SDLC lines. Specifies whether this line uses modems that are not LPDA2 modems but that support the LPDA2 modem dial facility.

Code on: GROUP, LINE (page 2-243).

LSPRI

```
      ┌─LSPRI=PU────┐
►►──┼─────────────┼──►◄
    └─LSPRI=─┬─PU───┬─┘
           ├─NO───┤
           ├─YES──┤
           └─LINK─┘
```
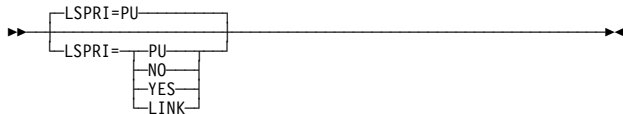
For frame-relay, ISDN, NTRI, and 3746 Model 900 Token-Ring lines in network control mode. Specifies whether transmission priority is to be used to determine the order in which PIUs are transmitted to the physical unit.

Code on: GROUP (page 2-125).

LSPRI

```
      ┌─LSPRI=PU────┐
►►──┼─────────────┼──►◄
    └─LSPRI=─┬─PU───┬─┘
           ├─NO───┤
           └─LINK─┘
```
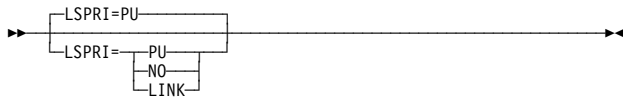
For frame-relay, ISDN, and 3746 Model 900 Token-Ring lines in network control mode. Specifies whether transmission priority is to be used to determine the order in which PIUs are transmitted to the physical unit.
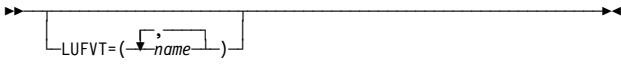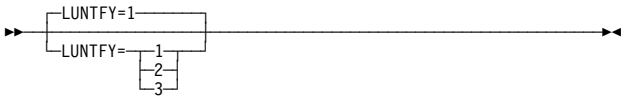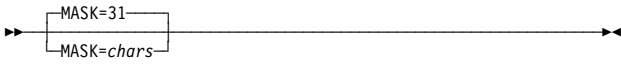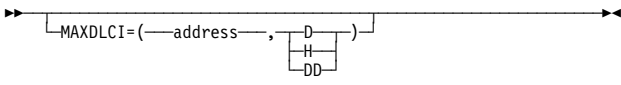
Code on: LINE (page 2-244).

LTRACE
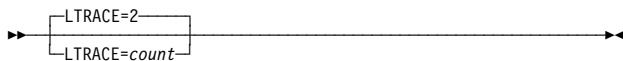
```
      ┌─LTRACE=2──────┐
►►──┼───────────────┼──►◄
    └─LTRACE=count─┘
```

For network control mode. Specifies the maximum number of line traces and scanner interface traces that NCP is to run concurrently.

Code on: BUILD (page 2-35).

Keyword	Description

LUCB

►►─┬──────────────────────┬─►◄
  └─LUCB=(─┬─;─────┬─name─)─┘
          └──────┘

For user-written code and Internet Protocol. Specifies the names of user-defined control blocks associated with a virtual logical unit.

Code on: GROUP, LINE, PU, LU (page 2-301).

LUDR

►►─┬──────────────┬─►◄
  └─LUDR=─┬─NO──┬─┘
         └─YES─┘

For dynamic reconfiguration. Specifies whether dynamic reconfiguration can delete this logical unit from the network.

Code on: GROUP, LINE, PU, LU (page 2-302).

LUFVT

►►─┬──────────────────────┬─►◄
  └─LUFVT=(─┬─;─────┬─name─)─┘
           └──────┘

For user-written code and Internet Protocol. Specifies the names of FVTs associated with a virtual logical unit.

Code on: GROUP, LINE, PU, LU (page 2-302).

LUNTFY

►►─┬─LUNTFY=1──────────┬─►◄
  └─LUNTFY=─┬─1─┬─────┘
           ├─2─┤
           └─3─┘

For user-written code. Specifies the use of the notify byte in user-written control blocks and whether control is to be passed to a user-written notify task.

Code on: GROUP, LINE, PU, LU (page 2-303).

MASK

►►─┬─MASK=31──────┬─►◄
  └─MASK=chars─┘

For TWX terminals. Specifies the hexadecimal mask character to be used during LCST entry selection.

Code on: MTALCST (page 2-321).

MAXBFRU

►►─┬─MAXBFRU=32────┬─►◄
  └─MAXBFRU=count─┘

Specifies how many buffer units the access method allocates for receiving data.

Code on: GROUP, LINE, PU (page 2-391).

MAXCOLL

►►─┬──────────────────┬─►◄
  └─MAXCOLL=count─┘

Specifies the maximum number of resources on which data can be collected at one time by NPM.

Code on: GROUP, LINE, PU, LU (page 2-303).

MAXDATA

►►─┬──────────────┬─►◄
  └─MAXDATA=size─┘

Specifies the maximum number of bytes the physical unit can send or receive in one data transfer.

Code on: GROUP, LINE, PU (page 2-391).

MAXDLCI

►►─┬──────────────────────────────┬─►◄
  └─MAXDLCI=(──address──,─┬─D──┬─)─┘
                         ├─H──┤
                         └─DD─┘

For frame-relay physical lines, NTRI frame handler logical links, and ISDN D-channel lines. Specifies the largest DLCI that can be used on the subport or internet host.

Code on: GROUP, LINE (page 2-245), PU.

MAXPU

Keyword	Description

MAXFRAME
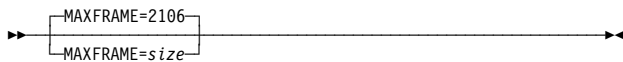
```
►►──┬─MAXFRAME=2106─┬──────────►◄
    └─MAXFRAME=size─┘
```

For frame-relay and ISDN physical lines and NTRI frame handler logical lines. Specifies the maximum size, in bytes, for frames transmitted over this physical or logical line.

Code on: GROUP, LINE (page 2-246).

MAXHELLO

```
►►──┬─MAXHELLO=6─────────┬──────►◄
    └─MAXHELLO=─┬─count─┬┘
               └─NONE──┘
```
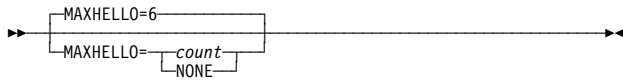
Specifies how many Hello datagrams NCP sends to NCPROUTE before a generic alert is issued.

Code on: IPOWNER (page 2-162).

MAXLEN

```
►►──┬───────────────┬──────────►◄
    └─MAXLEN=bytes──┘
```

Specifies the maximum size of the list of identifier sequences specified by the IDSEQ keyword.

Code on: IDLIST (page 2-150).

MAXLIST

```
►►──┬─────────────────┬────────►◄
    └─MAXLIST=entries─┘
```

Specifies the maximum number of entries in the SOT.

Code on: SERVICE (page 2-440).

MAXLU

```
►►──┬─────────────┬────────────►◄
    └─MAXLU=count─┘
```

For switched or dynamic reconfiguration lines under user-written line control. Specifies the maximum number of logical units that can be associated with the physical unit.

Code on: GROUP, LINE, PU (page 2-394).

MAXOUT

```
►►──┬────────────────┬─────────►◄
    └─MAXOUT=frames──┘
```

For nonswitched peripheral links. Specifies the maximum number of unacknowledged information frames that NCP is to send before requesting a response from the physical unit.

Also for nonswitched lines and a type 4 physical unit. Specifies the maximum number of information frames that NCP is to receive before requiring a response.

Code on: GROUP, LINE, PU (page 2-395).

MAXOUT

```
►►──┬─MAXOUT=7──────┬──────────►◄
    └─MAXOUT=frames─┘
```
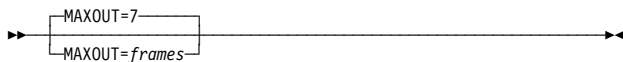
For secondary mode. Specifies the maximum number of unacknowledged information frames NCP is to send before a response is required from the adjacent NCP.

Code on: SDLCST (page 2-434).

MAXOUT

```
►►──┬─MAXOUT=7─────┬───────────►◄
    └─MAXOUT=count─┘
```

Specifies the maximum number of PIUs NCP is to send before requiring a response.
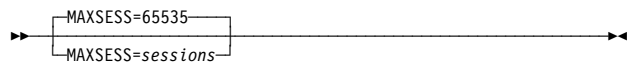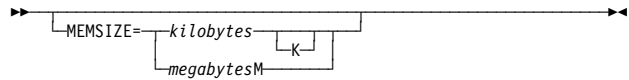
Code on: SWMMT (page 2-445).

MAXPU

```
►►──┬─────────────┬────────────►◄
    └─MAXPU=count─┘
```

For SDLC lines. Specifies the maximum number of physical units associated with this line.

Code on: GROUP, LINE (page 2-248).

Keyword	Description

MAXSESS

```
      ┌─MAXSESS=65535──┐
►►────┤                ├────►◄
      └─MAXSESS=sessions─┘
```

Specifies the maximum number of LU-LU sessions any single peripheral logical unit can have.

Code on: BUILD (page 2-36), GROUP, LINE, PU, LU (page 2-304).

MAXSSCP

```
►►──┬──────────────┬──────►◄
    └─MAXSSCP=sscps─┘
```

For network control mode. Specifies the maximum number of SSCPs with which NCP can be in session concurrently.

Code on: BUILD (page 2-36).

MAXSUBA

```
►►──┬─────────────────┬──────►◄
    └─MAXSUBA=address─┘
```

Specifies the upper limit of the range of subarea addresses used in the native network defined by this NETWORK definition statement.

Code on: BUILD (page 2-37), NETWORK (page 2-334).

MAXTP

```
►►──┬────────────┬──────►◄
    └─MAXTP=count─┘
```

Specifies the maximum number of resources on which data can be collected by transmission priority at one time by NPM. MAXTP is valid only for logical units defined within the NPM programmed resource group.

Code on: GROUP, LINE, PU, LU (page 2-304).

MAXTSL

```
►►──┬──────────────┬──────►◄
    └─MAXTSL=bytes─┘
```

For Token-Ring physical lines. Specifies the maximum amount of data NCP is to transmit over a Token-Ring connection.

Code on: GROUP, LINE (page 2-249).

MEMSIZE

```
►►──┬─────────────────────────────┬──────►◄
    │          ┌─kilobytes─┐      │
    └─MEMSIZE=─┤        ┌─K─┐     │
               │        └───┘     │
               └─megabytesM────────┘
```

Specifies the maximum amount of controller storage NCP or EP is to use, including storage for the load modules and buffers created after loading NCP into the communication controller.

Code on: BUILD (page 2-37).

METRIC

```
►►──METRIC=value────────────────────────►◄
```

Required for Internet Protocol. Specifies a value used by NCPROUTE to evaluate new route information received in RIP packets over the interface defined by the IPLOCAL definition statement.

Code on: IPLOCAL (page 2-157).

METRIC

```
►►──METRIC=value────────────────────────►◄
```

Required for Internet Protocol when you code the IPROUTE definition statement. For non-permanent routes, specifies a metric value used by NCPROUTE to evaluate new route information received in RIP packets. For permanent routes, specifies a metric value used by NCPROUTE to determine route selection order.

Code on: IPROUTE (page 2-168).

MODULO

Keyword	Description
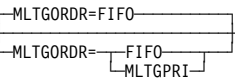
METRIC

```
►►──METRIC=value──────────────────────────────►◄
```

Frame-relay and channel internet resources, and internal NCP to Model 900 point-to-point IP connections. Specifies a value used by NCPROUTE to evaluate new route information received in RIP packets over the interface defined by the PU definition statement.

Code on: PU (page 2-399).

MLTGORDR

```
        ┌─MLTGORDR=FIFO─────┐
►►──────┤                   ├──────────────────►◄
        └─MLTGORDR=──┬─FIFO────┬─┘
                     └─MLTGPRI─┘
```

Specifies whether the individual lines in multilink transmission groups are to be selected for use in order of activation or in order of priority.

Code on: BUILD (page 2-38).

MLTGPRI

```
        ┌─MLTGPRI=0─────────┐
►►──────┤                   ├──────────────────►◄
        └─MLTGPRI=priority──┘
```

Assigns a priority value to a line in a multilink transmission group. Higher priority lines are selected before lower priority lines when MLTGORDR=MLTGPRI on the BUILD definition statement.

Code on: GROUP, LINE (page 2-250).

MODE

```
►►──┬────────────────────┬──────────────────────►◄
    └─MODE=──┬─PRImary───┬─┘
             └─SECondary─┘
```
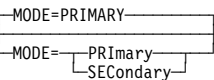
Specifies the mode of operation of a line group.

Code on: GROUP (page 2-127).

MODE

```
        ┌─MODE=PRIMARY──────┐
►►──────┤                   ├──────────────────►◄
        └─MODE=──┬─PRImary───┬─┘
                 └─SECondary─┘
```
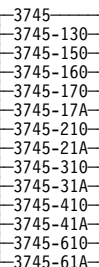
Specifies whether this selection table entry is to be used when the communication controller functions as a primary or secondary device.

Code on: SDLCST (page 2-435).

MODEL

```
►►──MODEL=──┬─3745─────┬──────────────────────►◄
            ├─3745-130─┤
            ├─3745-150─┤
            ├─3745-160─┤
            ├─3745-170─┤
            ├─3745-17A─┤
            ├─3745-210─┤
            ├─3745-21A─┤
            ├─3745-310─┤
            ├─3745-31A─┤
            ├─3745-410─┤
            ├─3745-41A─┤
            ├─3745-610─┤
            └─3745-61A─┘
```

Required. Specifies the IBM communication controller model for which this NCP or EP is to be generated.

Code on: BUILD (page 2-39).

MODEM

```
        ┌─MODEM=OPTION2─────┐
►►──────┤                   ├──────────────────►◄
        └─MODEM=──┬─OPTION2─┬─┘
                  ├─OPTION1─┤
                  └─NTT─────┘
```

For start-stop and BSC lines in emulation mode. Specifies whether this line is to be enabled or disabled after the communication controller is loaded (by IPL) or the System Reset key is pressed.

Code on: GROUP, LINE (page 2-250).

MODULO

```
        ┌─MODULO=8──────────┐
►►──────┤                   ├──────────────────►◄
        └─MODULO=──┬─8───┬──┘
                   └─128─┘
```

Specifies the maximum number of information frames NCP sends to a device before requiring a response.

Code on: GROUP, LINE, PU (page 2-399).

Keyword	Description

MONITOR

```
    ┌─MONITOR=NO──────────┐
►►──┤                     ├──►◄
    └─MONITOR=─┬─YES─┬────┘
              └─NO──┘
```
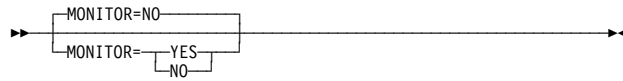
For the IBM 1050 and 2741, and for TWX and MTA lines in network control mode. Specifies whether NCP is to monitor this line for an attention signal or a disconnect condition at the terminal.

Code on: GROUP, LINE (page 2-251).

MONLINK

```
►►──┬─────────────────────────┬──►◄
    └─MONLINK=─┬─YES───────┬──┘
              ├─NO────────┤
              └─CONTinuous┘
```
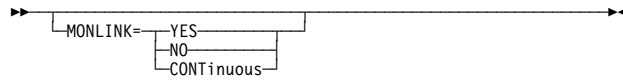
For SDLC subarea links. Specifies whether SMMF is to activate this link.

Code on: GROUP, LINE (page 2-252), PU (page 2-400).

MPTALT

```
    ┌─MPTALT=NO────────┐
►►──┤                  ├──►◄
    └─MPTALT=─┬─YES─┬──┘
             └─NO──┘
```

For switched call-out BSC lines in network control mode. Specifies whether this line is to be used as an alternate line for communicating with BSC stations normally reached over a nonswitched multipoint line.

Code on: GROUP, LINE (page 2-253).

MTALIST

```
►►──┬──────────────────┬──►◄
    └─MTALIST=name─────┘
```
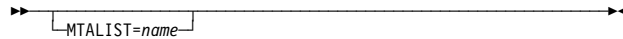
For call-in MTA lines in network control mode. Specifies that this line is used as a call-in MTA line and identifies the list of terminal types that can call the communication controller over this line.

Code on: GROUP, LINE (page 2-254).

MTARTO

```
    ┌─MTARTO=1.0──────┐
►►──┤                 ├──►◄
    └─MTARTO=seconds──┘
```

For MTA lines. Specifies the reply time-out interval to be used when a start-stop terminal on a switched MTA line calls the communication controller.

Code on: BUILD (page 2-39).

MTARTRY

```
    ┌─MTARTRY=0────────┐
►►──┤                  ├──►◄
    └─MTARTRY=retries──┘
```

For MTA lines. Specifies the maximum number of times NCP is to retry the start-stop MTA sign-on procedure after identifying the type of MTA terminal that called the communication controller.

Code on: BUILD (page 2-39).

MULTI

```
    ┌─MULTI=NO─────────┐
►►──┤                  ├──►◄
    └─MULTI=─┬─YES─┬───┘
            └─NO──┘
```

For IBM 2845 and 2848 Display Control units in emulation mode. Specifies whether this line connects the communication controller to one or more IBM 2845 or 2848 Display Control units equipped with multipoint line control, or to a single-display control equipped with point-to-point line control.

Code on: GROUP, LINE (page 2-254).

Keyword	Description

NAME

```
▶▶──┬────────────────────────┬──▶◀
    └─NAME=──name──────────┘
```

For predefined cross-network resource addresses. Specifies the resource name associated with the network address defined by this GWNAU definition statement.

Code on: GWNAU (page 2-145).

NAME

```
▶▶──NAME=name────────────────────▶◀
```

Required when you code the UBHR definition statement. Specifies the name of the user-written module that contains the block handling routine defined by this UBHR definition statement.

Code on: UBHR (page 2-475).

NAMTAB

```
▶▶──┬────────────────────┬──────▶◀
    └─NAMTAB=entries─────┘
```

Specifies how many entries the network names table is to contain.

Code on: BUILD (page 2-40).

NAUCB

```
                  ┌─,──────┐
▶▶──┬─────────────────────────┬──▶◀
    └─NAUCB=(──▼─name──┴─)─┘
```

Specifies the names of user-defined control blocks associated with the NAU defined by this NCPNAU definition statement.

Code on: NCPNAU (page 2-328).

NAUFVT

```
               ┌─,──────┐
▶▶──NAUFVT=(──▼─name──┴─)──────▶◀
```

Required when you code the NCPNAU definition statement. Specifies the names of the FVTs associated with the NAU defined by this NCPNAU definition statement.

Code on: NCPNAU (page 2-329).

NCPCA

```
      ┌─NCPCA=ACTIVE──────────┐
▶▶──┴──────────────────────────┴──▶◀
    └─NCPCA=──┬─ACTIVE──┬─┘
              └─INACTIVE─┘
```

For channel links. Specifies whether the channel adapter is active or inactive.

Code on: GROUP, LINE (page 2-254).

NCPTRACE

```
      ┌─NCPTRACE=OFF──────────┐
▶▶──┴──────────────────────────┴──▶◀
    └─NCPTRACE=──┬─ON──┬──┘
                 └─OFF─┘
```

For frame relay and NTRI. Specifies whether the snap trace function is to be enabled for frame-relay and NTRI resources.

Code on: BUILD (page 2-40).

NCST

```
▶▶──┬────────────┬───────────────▶◀
    └─NCST=IP───┘
```

For Internet Protocol. Specifies that the logical units in this line group support Internet Protocol using NCST.
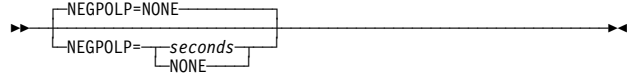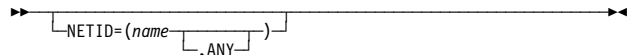
Code on: GROUP (page 2-127).

NDNAME

```
▶▶──┬────────────────┬───────────▶◀
    └─NDNAME=name───┘
```

Supports the automatic recataloging of the NEWDEFN file when NDF is running under VSE.

Code on: OPTIONS (page 2-343).

Keyword	Description

NEGPOLP

```
          ┌─NEGPOLP=NONE────────┐
►►────────┤                     ├──────────────►◄
          └─NEGPOLP=─┬─seconds─┬┘
                     └─NONE────┘
```

For multipoint BSC lines in network control mode. Specifies how long NCP is to pause before resuming polling after receiving a negative response to polling from a multipoint BSC station.

Code on: GROUP, LINE (page 2-255).

NERLIM

```
►►──┬────────────────┬──────────────────────────►◄
    └─NERLIM=─┬─8──┬─┘
             └─16─┘
```

For the migration aid function. Specifies that the migration aid is to add the ERLIMIT keyword to any NETWORK definition statement on which it is not already coded, and specifies the value for it.

Code on: OPTIONS (page 2-344).

NETID

```
►►──NETID=name──────────────────────────────────►◄
```

Required. Specifies the name of the native network.

Code on: BUILD (page 2-41).

NETID

```
►►──┬──────────────────────────┬────────────────►◄
    └─NETID=(name─┬──────┬─)────┘
                  └─,ANY─┘
```

For a gateway NCP. Specifies the name of the network associated with the subarea line group you are defining. Also specifies whether NCP can negotiate the name of the adjacent network during link activation.

Code on: GROUP (page 2-127).

NETID

```
►►──┬─────────────┬──────────────────────────────►◄
    └─NETID=name──┘
```

For predefined cross-network resource addresses. Specifies a network identifier to be associated with a resource address.

Code on: GWNAU (page 2-146).

NETID

```
►►──┬─────────────┬──────────────────────────────►◄
    └─NETID=name──┘
```

For a gateway NCP. Specifies the name of the network associated with an adjacent link station.

Code on: LINE, PU (page 2-400).

NETID

```
►►──┬─────────────┬──────────────────────────────►◄
    └─NETID=name──┘
```

For independent logical units. Specifies the name of the network that contains the logical unit you are defining.

Code on: LU (page 2-305).

NETID

```
►►──┬─────────────┬──────────────────────────────►◄
    └─NETID=name──┘
```

Required when you code the NETWORK definition statement unless you code the COPIES keyword on the NETWORK definition statement. Specifies the name of this non-native network.

Code on: NETWORK (page 2-335).

NETID

```
►►──┬─────────────┬──────────────────────────────►◄
    └─NETID=name──┘
```

Specifies the network name of the designated caller.

Code on: SWMMT (page 2-445).

NEXTADDR

Keyword	Description

NETLIM

```
►►─┬────────────────┬─►◄
   └─NETLIM=hscbs──┘
```

For a gateway NCP. Specifies how many HSCBs can be assigned at any one time to all addresses in the native or non-native network.

Code on: BUILD (page 2-41), NETWORK (page 2-335).

NETWORK

```
    ┌─NETWORK=SNA──────┐
►►─┼──────────────────┼─►◄
   └─NETWORK=─┬─SNA──┬─┘
             ├─IP───┤
             └─FRELAY─┘
```
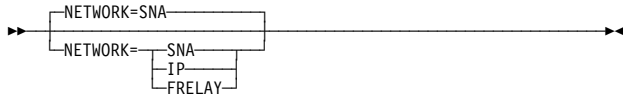
Specifies the type of network this physical unit represents to the Token-Ring physical line.

Code on: PU (page 2-401).

NEWDEFN

```
►►─┬──────────────────────────────────────────────────┬─►◄
   └─NEWDEFN=─┬─NO────────────────────────────────┬──┘
             └─(YES,┬─ECHO─┬,┬─NOSUPP─┬,┬─REUSE───┬)─┘
                    ├─ECHO─┤ └─SUPP───┘ ├─REUSE───┤
                    └─PACK─┘            └─NOREUSE─┘
```
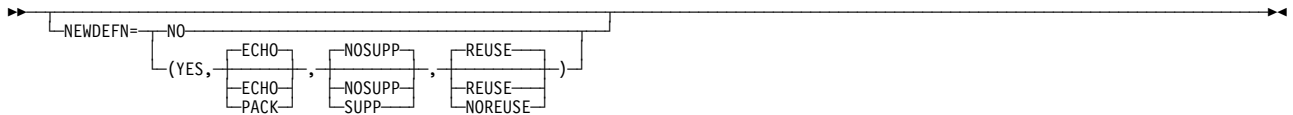
Specifies whether NDF is to create a NEWDEFN file, and specifies various characteristics of the NEWDEFN file.

Code on: OPTIONS (page 2-344).

NEWNAME

```
►►─┬──────────────────┬─►◄
   └─NEWNAME=name────┘
```

Defines the name of the NCP load module. Also specifies the physical unit name if the PUNAME keyword is not defined on the PU definition statement.

Code on: BUILD (page 2-41).

NEWSYNC

```
    ┌─NEWSYNC=NO──┐
►►─┼─────────────┼─►◄
   └─NEWSYNC=─┬─YES─┬─┘
             └─NO──┘
```

For BSC and SDLC lines. Specifies whether the communication controller is to supply the new-sync signal to the modem on this line.

Code on: GROUP, LINE (page 2-255).

NEXT
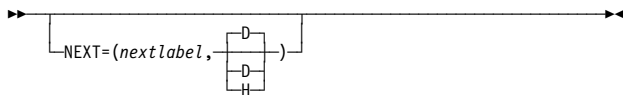
```
►►─┬──────────────────────────┬─►◄
   └─NEXT=(nextlabel,┬─D─┬)──┘
                     ├─D─┤
                     └─H─┘
```

Specifies a convention for constructing unique labels for the definition statements that will be copied when you code AUTOCOPY on the LINE definition statement.
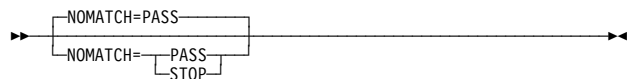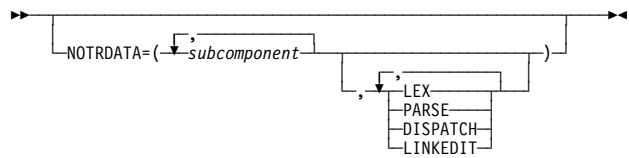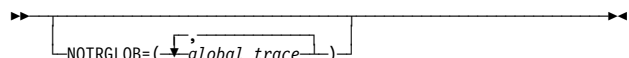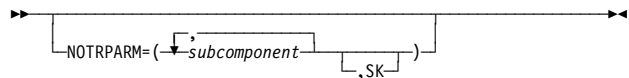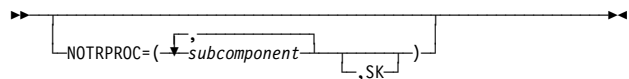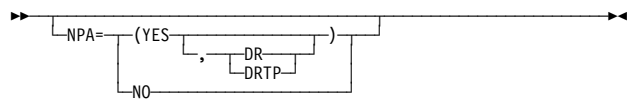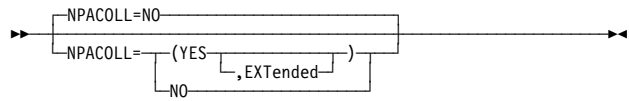
Code on: CLUSTER (page 2-72), LU (page 2-305), PU (page 2-401), TERMINAL (page 2-466), COMP.

NEXTADDR

```
    ┌─NEXTADDR=0───────┐
►►─┼──────────────────┼─►◄
   └─NEXTADDR=address─┘
```

For Internet Protocol. Specifies the internet address of the next internet gateway to which an IP datagram is to be routed on its way to the destination network or subnetwork, or the internet address of the NCST logical unit session partner in a point-to-point connection.

Code on: IPROUTE (page 2-168).

Keyword	Description

NOMATCH

```
        ┌─NOMATCH=PASS─┐
►►──────┤              ├──────────────────────►◄
        └─NOMATCH─┬─PASS─┬─┘
                  └─STOP─┘
```

Specifies the action NCP is to perform if it does not receive an identifier sequence from a calling device or if it receives an identifier sequence not defined by this IDLIST definition statement.

Code on: IDLIST (page 2-151).

NOTIFY

```
        ┌─NOTIFY=1─┐
►►──────┤          ├──────────────────────────►◄
        └─NOTIFY=─┬─1─┬─┘
                  └─2─┘
```

Specifies how the notify byte in user-written control blocks is to be used and whether control is to be passed to a user-written notify task.

Code on: NCPNAU (page 2-329).

NOTRDATA

```
►►──┬──────────────────────────────────────┬──►◄
    │              ┌─,◄───────────┐        │
    └─NOTRDATA=(─┬─▼─subcomponent─┴─┐──────)
                 │    ┌─,◄──────┐   │
                 └────▼─┬─LEX──────┬┴─┘
                        ├─PARSE────┤
                        ├─DISPATCH─┤
                        └─LINKEDIT─┘
```

Specifies the subcomponent or data areas for which the data trace is to be deactivated.

Code on: OPTIONS (page 2-345).

NOTRGLOB

```
►►──┬─────────────────────────────────┬───────►◄
    │             ┌─,◄───────────┐    │
    └─NOTRGLOB=(─▼─global trace─┴─)─┘
```

Deactivates global trace activity for specified global traces.

Code on: OPTIONS (page 2-346).

NOTRPARM

```
►►──┬──────────────────────────────────┬──────►◄
    │             ┌─,◄──────────┐       │
    └─NOTRPARM=(─▼─subcomponent─┴─┬─────)
                                  └─,SK─┘
```

Specifies the subcomponent or statement and keyword routines for which the I/O parameter trace is to be deactivated.

Code on: OPTIONS (page 2-346).

NOTRPROC

```
►►──┬──────────────────────────────────┬──────►◄
    │             ┌─,◄──────────┐       │
    └─NOTRPROC=(─▼─subcomponent─┴─┬─────)
                                  └─,SK─┘
```

Specifies the subcomponent or statement and keyword routines for which the procedure entry and exit trace is to be deactivated.

Code on: OPTIONS (page 2-347).

NPA

```
►►──┬───────────────────────────────┬─────────►◄
    └─NPA=─┬─(YES─┬──────┬─)─┬─────┘
           │      ├─DR───┤   │
           │      └─DRTP─┘   │
           └─NO──────────────┘
```

Specifies whether NCP is to include the network performance analyzer function.

Code on: BUILD (page 2-42).

NPACOLL

```
        ┌─NPACOLL=NO─────────────────┐
►►──────┤                            ├──────────►◄
        └─NPACOLL=─┬─(YES─┬───────────┬─)─┘
                   │      └─,EXTended─┘
                   └─NO──────────────────┘
```

Specifies whether performance data for this terminal is eligible for collection by the network performance analyzer and, if so, whether the form of this data is different from that used by NCP.

Code on: CLUSTER (page 2-73), TERMINAL (page 2-468).

Keyword	Description

NPACOLL

```
▶▶──┬─────────────────────────────────────┬──▶◀
    └─NPACOLL=─┬─(YES─┬──────────────┬─)─┬─┘
              │      └─,EXTended─────┘   │
              └─NO──────────────────────┘
```
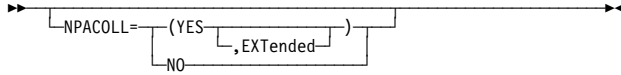
Specifies whether performance data for this line is eligible for collection by the network performance analyzer and, if so, whether the form of this data is different from that used by NCP.

Code on: GROUP, LINE (page 2-256).

NPACOLL
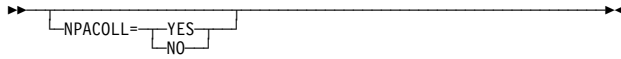
```
▶▶──┬─────────────────────┬──▶◀
    └─NPACOLL=─┬─YES─┬─────┘
              └─NO──┘
```

Specifies whether performance data for this logical unit is eligible for collection by the network performance analyzer.

Code on: LU (page 2-306).

NPACOLL

```
▶▶──┬─────────────────────────────────────┬──▶◀
    └─NPACOLL=─┬─(YES─┬──────────────┬─)─┬─┘
              │      └─,EXTended─────┘   │
              └─NO──────────────────────┘
```
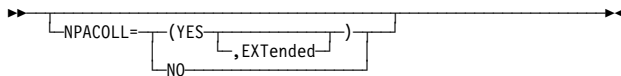
Specifies whether performance data for this physical unit is eligible for collection by the network performance analyzer.

Code on: PU (page 2-403).

NPARSC
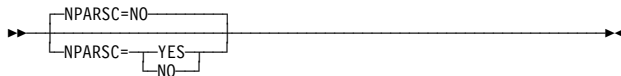
```
    ┌─NPARSC=NO────────┐
▶▶──┼──────────────────┼──▶◀
    └─NPARSC=─┬─YES─┬───┘
             └─NO──┘
```

Specifies that this line group is a network performance analyzer resource used to send and receive NPM data and commands between NCP and the host.

Code on: GROUP (page 2-128).

NPATP

```
    ┌─NPATP=NO────────┐
▶▶──┼─────────────────┼──▶◀
    └─NPATP=─┬─YES─┬───┘
            └─NO──┘
```

For SDLC resources. Specifies whether network performance data is to be collected by PIU transmission priority on the line or physical unit in addition to basic network performance data.

Code on: GROUP, LINE (page 2-257), PU (page 2-404).

NRZI

```
    ┌─NRZI=YES────────┐
▶▶──┼─────────────────┼──▶◀
    └─NRZI=─┬─YES─┬────┘
           └─NO──┘
```
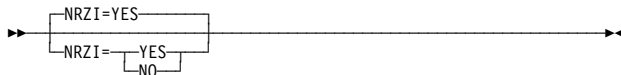
For SDLC lines in network control mode. Specifies whether the DTE at the end of the SDLC link must operate in NRZI mode or in NRZ mode.

Code on: GROUP, LINE (page 2-258).

NTUNECOL

```
    ┌─NTUNECOL=NO────────┐
▶▶──┼────────────────────┼──▶◀
    └─NTUNECOL=─┬─YES─┬───┘
               └─NO──┘
```
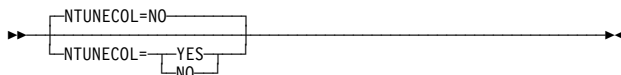
Specifies whether collection in the NCP is active for NTune. NCP collects data for the following:

- 3745-attached SDLC lines and stations
- 3745-attached channels
- 3745 frame-relay physical stations
- 3745 frame-relay logical lines
- NTRI physical lines
- NTRI physical stations
- NTRI frame handler logical lines
- NTRI logical FHSP stations.

Code on: BUILD (page 2-43)

Keyword	Description

NUMADDR

```
►►──┬──────────────────────┬──►◄
    └─NUMADDR=number─┘
```

For a pool of reusable cross-network resources. Specifies the number of addresses reserved for cross-network resource address assignment.

Code on: GWNAU (page 2-146).

NUMBER

```
►►──NUMBER=count──────────────►◄
```

Required when you code the LUPOOL definition statement. Specifies the number of logical units to be included in the dynamic reconfiguration logical unit pool.

Code on: LUPOOL (page 2-315).

NUMBER

```
►►──┬─NUMBER=0─────┬──►◄
    └─NUMBER=count─┘
```

Specifies the number of physical units to be included in the dynamic reconfiguration physical unit pool.

Code on: PUDRPOOL (page 2-430).

NUMDRIF

```
►►──┬─NUMDRIF=10──┬──►◄
    └─NUMDRIF=nnn─┘
```

Specifies the number of frame-relay internet interfaces that can be dynamically added to the network.

Code on: IPOWNER (page 2-162).

NUMHSAS

```
►►──┬─NUMHSAS=128──────┬──►◄
    └─NUMHSAS=subareas─┘
```

Specifies the number of subareas in the native with virtual routes ending in this NCP.

Code on: BUILD (page 2-44).

NUMHSAS

```
►►──┬──────────────────────┬──►◄
    └─NUMHSAS=subareas─┘
```

Specifies the number of subareas in the non-native network with virtual routes ending in this NCP.

Code on: NETWORK (page 2-336).

NUMILU

```
►►──┬─NUMILU=0─────┬──►◄
    └─NUMILU=count─┘
```

Specifies the number of independent logical units to be included in the dynamic reconfiguration logical unit pool.

Code on: LUDRPOOL (page 2-311).

NUMROUTE

```
►►──┬─NUMROUTE=(25,25,25)──────────────┬──►◄
    │              ┌─25─┐  ┌─25─┐  ┌─25─┐ │
    └─NUMROUTE=(──┴─n1─┴─,─┴─n2─┴─,─┴─n3─┴──)─┘
```

Specifies how many additional internet route table entries are to be made available for management by the NCPROUTE program.

Code on: IPOWNER (page 2-163).

NUMSESS

```
►►──┬──────────────────┬──►◄
    └─NUMSESS=hscbs─┘
```

For user-written code. Specifies the number of HSCBs to be associated with the NAU control block.

Code on: GROUP, LINE, PU, LU (page 2-307).

NUMSESS

```
►►──┬─NUMSESS=1──────┬──►◄
    └─NUMSESS=number─┘
```

For predefined cross-network resource addresses. Specifies the number of HSCBs to be assigned permanently to the address being defined.

Code on: GWNAU (page 2-146).

ORDLO

Keyword	Description

NUMSESS

```
►►──┬─────────────────┬──►◄
     └─NUMSESS=hscbs──┘
```

Specifies how many HSCBs are to be associated with the resource.

Code on: NCPNAU (page 2-329).

NUMTYP1

```
      ┌─NUMTYP1=0──────┐
►►──┼────────────────┼──►◄
      └─NUMTYP1=count──┘
```

Specifies how many logical units the dynamic reconfiguration pool is to contain for type 1 physical units.

Code on: LUDRPOOL (page 2-312).

NUMTYP2

```
      ┌─NUMTYP2=0──────┐
►►──┼────────────────┼──►◄
      └─NUMTYP2=count──┘
```

Specifies how many logical units the dynamic reconfiguration pool is to contain for type 2 physical units.

Code on: LUDRPOOL (page 2-313).

OLT

```
      ┌─OLT=YES────────┐
►►──┼────────────────┼──►◄
      └─OLT=─┬─YES─┬──┘
             └─NO──┘
```

For network control mode. Specifies whether NCP is to include the optional online terminal test and online line test facilities.

Code on: BUILD (page 2-45).

OPTIONS

```
              ┌─,────────┐
►►──OPTIONS=(─▼─facility─┴─)──────►◄
```

Required. Specifies which dynamic control facilities NCP is to include.

Code on: SYSCNTRL (page 2-446).

ORDER

```
      ┌───────────────────┐
      │         ┌─,──────┐ │
►►──┴─ORDER=(─▼─device─┴─)─┴──►◄
```

Specifies the order in which the devices on this line are to be serviced.

Code on: SERVICE (page 2-440).

ORDHI

```
      ┌────────────────────┐
      │        ┌─,─────┐    │
►►──┴─ORDHI=(─▼─name─┴─)──┴──►◄
```

Specifies the macro library members that contain linkage editor ORDER statements for object modules that are not level 2 or level 3 code. These object modules may reside anywhere in communication controller storage.

Code on: GENEND (page 2-98).

ORDINIT

```
      ┌──────────────────────┐
      │          ┌─,─────┐    │
►►──┴─ORDINIT=(─▼─name─┴─)──┴──►◄
```

Specifies the macro library members that contain linkage editor ORDER statements for user-written initialization code.

Code on: GENEND (page 2-99).

ORDLO

```
      ┌───────────────────┐
      │        ┌─,─────┐   │
►►──┴─ORDLO=(─▼─name─┴─)──┴──►◄
```

Specifies the macro library members that contain linkage editor ORDER statements for object modules that are not level 2 or level 3 code. These object modules will reside in the lower 64KB of communication controller storage.

Code on: GENEND (page 2-98).

Keyword	Description

ORDL2HI

```
►►─┬─────────────────────────┬─►◄
   └─ORDL2HI=(─┬─────┬─name─)─┘
              └─,◄──┘
```

Specifies the macro library members that contain linkage editor ORDER statements for object modules that are level 2 or level 3 code. These object modules may reside anywhere in communication controller storage.

Code on: GENEND (page 2-99).

ORDL2LO

```
►►─┬─────────────────────────┬─►◄
   └─ORDL2LO=(─┬─────┬─name─)─┘
              └─,◄──┘
```

Specifies the macro library members that contain linkage editor ORDER statements for object modules that are level 2 or level 3 code. These object modules will reside in the lower 64KB of communication controller storage.

Code on: GENEND (page 2-99).

OWNERPL

```
      ┌─OWNERPL=NONE───────────────────┐
►►────┼────────────────────────────────┼─►◄
      └─OWNERPL=─┬─NONE─────────────┬───┘
                 ├─ALL──────────────┤
                 └─(─┬─,◄──┬─TR─────┬─)─┘
                              ├─ODLCTR─┤
                              ├─FR─────┤
                              ├─ODLCFR─┤
                              └─ISDN───┘
```
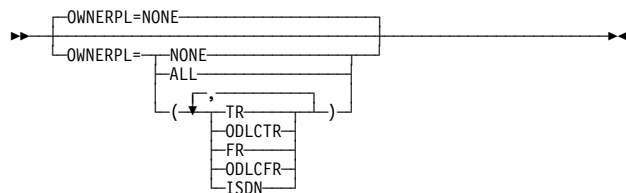
Specifies the types of unassigned logical resources that are managed in owner-based pools.

Code on: BUILD (page 2-45).

PACING

```
      ┌─PACING=(1,,ADAP)──────────────────────┐
►►────┼───────────────────────────────────────┼─►◄
      └─PACING=(─┬─1──────┬─,─┬──────┬─,─┬─ADAP──┬─)─┘
                 └─number─┘   └─vtam─┘   ├─ADAP──┤
                                         └─FIXED─┘
```
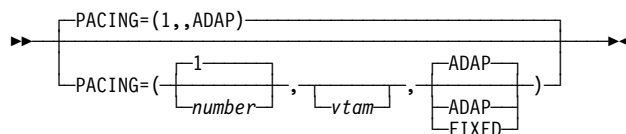
Overrides the fixed pacing window size or the adaptive pacing batch mode window size for this logical unit.

Code on: GROUP, LINE, PU, LU (page 2-307).

PAD
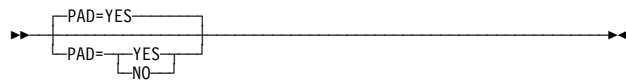
```
      ┌─PAD=YES───────┐
►►────┼───────────────┼─►◄
      └─PAD=─┬─YES─┬──┘
             └─NO──┘
```

For BSC lines in emulation mode. Specifies whether NCP, when emulating IBM 2703 Transmission Control, verifies the first 4 bits of trailing pad characters received from this line.

Code on: GROUP, LINE (page 2-259).

PADCNT

```
►►─┬──────────────────┬─►◄
   └─PADCNT=count──────┘
```

For WTTY lines in network control mode. Specifies the number of idle characters to be sent to a WTTY terminal to permit its motor to reach full speed before receiving data.

Code on: GROUP (page 2-128).

PARCHK
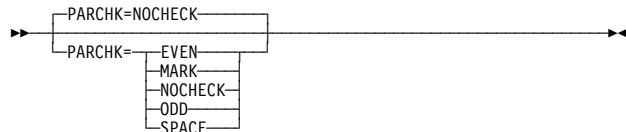
```
      ┌─PARCHK=NOCHECK────┐
►►────┼───────────────────┼─►◄
      └─PARCHK=─┬─EVEN────┬┘
                ├─MARK────┤
                ├─NOCHECK─┤
                ├─ODD─────┤
                └─SPACE───┘
```

For TWX lines in network control mode. Specifies whether NCP is to set odd, even, mark, or space parity or is not to set parity

Code on: GROUP, LINE (page 2-259).

PARGEN
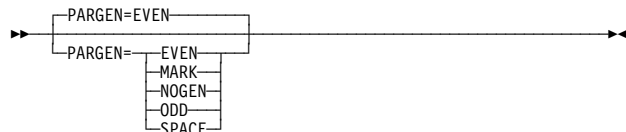
```
      ┌─PARGEN=EVEN───────┐
►►────┼───────────────────┼─►◄
      └─PARGEN=─┬─EVEN───┬─┘
                ├─MARK───┤
                ├─NOGEN──┤
                ├─ODD────┤
                └─SPACE──┘
```

For TWX lines in network control mode. Specifies whether NCP is to set odd, even, mark, or space parity or is not to set parity for data transmitted on this line.

Code on: GROUP, LINE (page 2-259).

PIUCHECK

Keyword	**Description**

PASSLIM

```
►►─┬──────────────────┬─►◄
   └─PASSLIM=count─────┘
```

For nonswitched links. Specifies the maximum number of consecutive PIUs or PIU segments that NCP is to send at one time to this physical unit.

Code on: GROUP, LINE, PU (page 2-404).

PASSLIM
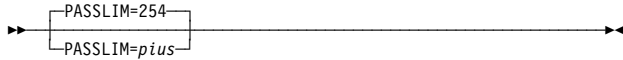
```
    ┌─PASSLIM=254───┐
►►──┼───────────────┼─►◄
    └─PASSLIM=pius──┘
```

Specifies the maximum number of consecutive PIUs or PIU segments that NCP is to send at one time to the adjacent NCP.

Code on: SDLCST (page 2-435).

PATHEXT

```
►►─┬──────────────────┬─►◄
   └─PATHEXT=entries──┘
```

For dynamic path update facility. Specifies how many extra TRT entries for the native and non-native network are to be available for added destination subareas.

Code on: BUILD (page 2-46), NETWORK (page 2-336).

PAUSE

```
    ┌─PAUSE=(0.2,2.8)─────────────────┐
►►──┼─────────────────────────────────┼─►◄
    │         ┌─0.2─┐   ┌─2.8─┐        │
    └─PAUSE=(─┼─────┼─,─┼─────┼─)──────┘
              └─t1──┘   └─t2──┘
```
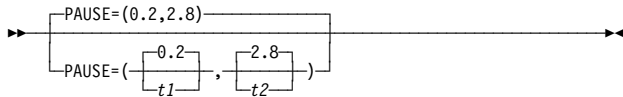
Specifies the delay or cycle time interval according to the link value.

Code on: GROUP, LINE (page 2-260).

PAUSE

```
►►─┬────────────────────────────────────────┬─►◄
   └─PAUSE=(─┬───────────┬─┬──────────────┬─)┘
            └─modulo-8──┘ └─,modulo-128──┘
```
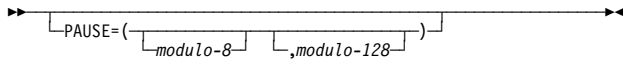
Specifies the average duration of the polling cycle time for SDLC primary or secondary subarea links.

Code on: SDLCST (page 2-435).

PECHAR

```
    ┌─PECHAR=FF────────┐
►►──┼──────────────────┼─►◄
    └─PECHAR=─┬─FF───┬──┘
             └─char─┘
```

For TWX lines in network control mode. Specifies the hexadecimal representation of the data character NCP uses to overlay data received from a TWX terminal when NCP detects a parity error.

Code on: GROUP (page 2-129).

PHYPORT

```
►►─┬─────────────────────────┬─►◄
   └─PHYPORT=─┬─number─┬──────┘
            └─NONE───┘
```

For NTRI logical lines. Associates the logical lines in a group with a physical line.

Code on: GROUP (page 2-129).

PHYSRSC
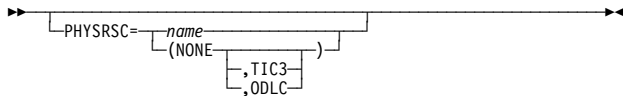
```
►►─┬──────────────────────────────────┬─►◄
   └─PHYSRSC=─┬─name──────────────┬────┘
            └─(NONE─┬────────┬─)──┘
                    ├─,TIC3──┤
                    └─,ODLC──┘
```

For Token-Ring, frame-relay, ISDN, and ESCON logical lines. Associates the logical lines in a group with the link station for a particular physical line.

Code on: GROUP (page 2-130).
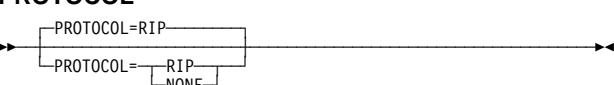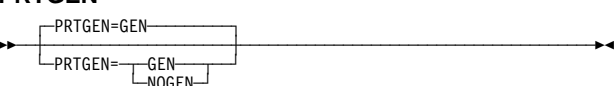
| **PIUCHECK**
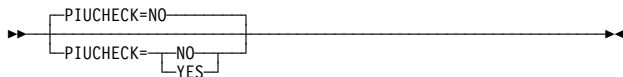
```
    ┌─PIUCHECK=NO──────┐
►►──┼──────────────────┼─►◄
    └─PIUCHECK=─┬─NO──┬─┘
              └─YES─┘
```

For frame-relay logical subarea stations, ISDN logical subarea stations, SDLC subarea stations, and Token-Ring logical subarea stations. Specifies whether a checksum validation is performed on PIUs sent and received by this station.

Code on: PU (page 2-408).

Keyword	Description

POLIMIT

```
         ┌─POLIMIT=(1,NOWAIT)──────────┐
►►───────┤                             ├──────►◄
         │           ┌─1─┐   ┌─NOWAIT─┐│
         └─POLIMIT=(──┤   ├─,─┤        ├)┘
                      └count┘  ├─WAIT───┤
                               ├─NOWAIT─┤
                               └─QUEUE──┘
```

For nonswitched multipoint start-stop or BSC lines in network control mode. Specifies the number of consecutive negative responses to polling that NCP is to accept from a device and the action that NCP is to take when that number is reached.

Code on: GROUP, LINE (page 2-261).

POLL

```
          ┌──,───┐
►►──POLL=(─▼─chars─┴─)──────────────────►◄
```

Required when you code the MTAPOLL definition statement. Specifies, in hexadecimal, the EBCDIC polling characters for all IBM 1050 terminals that can dial the communication controller over an MTA link.

Code on: MTAPOLL (page 2-325).

POLL

```
►►─────────────────────────────────────►◄
   └─POLL=chars─┘
```

Specifies, in hexadecimal, the EBCDIC polling characters assigned to the device.

Code on: TERMINAL (page 2-468), COMP.

POLLED

```
   ┌─POLLED=NO──┐
►►─┤            ├───────────────────────►◄
   └─POLLED=─┬─YES─┬─┘
             └─NO──┘
```

For start-stop and BSC lines in network control mode. Specifies whether devices attached to this line must be polled and addressed.

Code on: GROUP, LINE (page 2-262).

POLLTO

```
   ┌─POLLTO=ERROR──┐
►►─┤               ├────────────────────►◄
   └─POLLTO=─┬─ERROR───┬─┘
             └─NEGRESP─┘
```

For start-stop lines in network control mode. Specifies whether a time-out occurring during polling is to be treated as an error condition or a negative response.

Code on: GROUP, LINE (page 2-263).

PORT

```
►►─────────────────────────────────────►◄
   └─PORT=─┬─A─┬─┘
           ├─B─┤
           ├─C─┤
           └─D─┘
```

For SDLC or BSC lines attached to a local IBM 786x modem with the data multiplexer feature. Specifies the port in the local IBM 786x modem to which this line is connected.

Code on: LINE (page 2-263).

PORTADD

```
►►─────────────────────────────────────►◄
   └─PORTADD=number─┘
```

For Token-Ring, frame-relay, and ISDN physical lines. Assigns a number to the physical line to identify the associated TIC to VTAM.

Code on: LINE (page 2-263).

PRODID

```
►►─────────────────────────────────────►◄
   └─PRODID=name─┘
```

For IBM special products or user-written code. Specifies the name of the product identifier (subvector X'11') defined for the IBM special product or user-written code.

Code on: GROUP (page 2-131).

Keyword	Description

PROMPT

```
          ┌─PROMPT=YES─┐
►►────────┼────────────┼────────────────►◄
          └─PROMPT=─┬─YES─┬─┘
                    └─NO──┘
```

For TWX lines in network control mode. Specifies whether, at READ time, NCP is to transmit the prompt defined by TWXID on the BUILD definition statement.

Code on: GROUP, LINE (page 2-265).

PROTOCOL

```
          ┌─PROTOCOL=RIP─┐
►►────────┼──────────────┼──────────────►◄
          └─PROTOCOL=─┬─RIP──┬─┘
                      └─NONE─┘
```

For Internet Protocol. Specifies that the Routing Information Protocol (RIP) can manage information about the router interface specified on the IPLOCAL definition statement.

Code on: IPLOCAL (page 2-158).

PROTOCOL

```
          ┌─PROTOCOL=RIP─┐
►►────────┼──────────────┼──────────────►◄
          └─PROTOCOL=─┬─RIP──┬─┘
                      └─NONE─┘
```

Frame-relay and channel internet resources, and internal NCP to Model 900 point-to-point IP connections. Specifies that the Routing Information Protocol (RIP) can manage information about the router interface specified on the PU definition statement.

Code on: PU (page 2-408).

PRTGEN

```
          ┌─PRTGEN=GEN─┐
►►────────┼────────────┼────────────────►◄
          └─PRTGEN=─┬─GEN───┬─┘
                    └─NOGEN─┘
```

Specifies whether macro-generated statements are to be printed for the NCP table assemblies.

Code on: BUILD (page 2-46).

PT1

```
►►──┬─────────────┬──────────────────────►◄
    └─PT1=name────┘
```

For block handling routines. Specifies the name of the block handler that is to run when a request for the device is received from the host processor before a line is available.

Code on: BHSET (page 2-5).

PT2

```
►►──┬─────────────┬──────────────────────►◄
    └─PT2=name────┘
```

For block handling routines. Specifies the name of the block handler that is to run (1) when a request for the device is received from the host processor or (2) when data is received from the device while a line is available.

Code on: BHSET (page 2-5).

PT2EXEC

```
          ┌─PT2EXEC=BEFORE─┐
►►────────┼────────────────┼────────────►◄
          └─PT2EXEC=─┬─BEFORE─┬─┘
                     └─AFTER──┘
```

For block handling routines. Specifies whether this user-written routine is to process outgoing message data before or after the I/O operation.

Code on: DATETIME (page 2-77), UBHR (page 2-476).

PT3

```
►►──┬─────────────┬──────────────────────►◄
    └─PT3=name────┘
```

For block handling routines. Specifies the name of the block handler that is to run when data is received from the device after a line is no longer available.

Code on: BHSET (page 2-5).

Keyword	Description

PT3EXEC

```
        ┌─PT3EXEC=NO─────────┐
►►──────┤                    ├──────►◄
        └─PT3EXEC=─┬─YES─┐───┘
                   └─NO──┘
```

For block handling routines. Specifies whether a block handler set that runs at point 3 is to be associated with this station.

Code on: GROUP, LINE, CLUSTER (page 2-74), TERMINAL (page 2-469), COMP.

PUCB

```
►►──┬──────────────┬──────►◄
    └─PUCB=name────┘
```

For user-written code and Internet Protocol. Specifies the name of the user-defined control block to be associated with this virtual physical unit.

Code on: GROUP, LINE, PU (page 2-408).

PUDR

```
►►──┬───────────────┬──────►◄
    └─PUDR=─┬─NO──┐─┘
            └─YES─┘
```

For dynamic reconfiguration on a nonswitched link. Specifies whether the physical unit can be deleted from the network using the dynamic reconfiguration function.

Code on: GROUP, LINE, PU (page 2-408).

PUFVT

```
►►──┬───────────────┬──────►◄
    └─PUFVT=name────┘
```

For user-written code. Specifies the FVT name associated with a virtual physical unit.

Code on: GROUP, LINE, PU (page 2-410).

PUNAME

```
►►──┬────────────────┬──────►◄
    └─PUNAME=name────┘
```

Specifies the physical unit name for this NCP.

Code on: BUILD (page 2-46).

PUNTFY

```
►►──┬──────────────┬──────►◄
    └─PUNTFY=─┬─1─┐─┘
             ├─2─┤
             └─3─┘
```
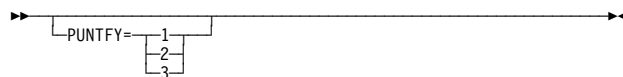
For user-written code and Internet Protocol. Defines the use of the notify byte in user-written control blocks and specifies whether control is to be passed to a user-written notify task.

Code on: GROUP, LINE, PU (page 2-410).

PUTYPE

```
►►──┬──────────────────┬──────►◄
    └─PUTYPE=─┬─1─────┐─┘
             ├─2─────┤
             ├─(1,2)─┤
             ├─4─────┤
             └─5─────┘
```
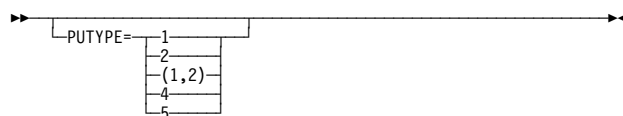
Specifies the physical unit type of the SDLC device being defined.

Code on: GROUP (page 2-131), LINE, PU (page 2-411).

PWROFF

```
        ┌─PWROFF=NO──────────┐
►►──────┤                    ├──────►◄
        └─PWROFF=─┬─YES─┐────┘
                  └─NO──┘
```

For a channel-attached NCP or PEP. Specifies whether NCP in a channel-attached communication controller is to turn off the power of a link-attached communication controller when it receives a command from the channel-attached host.

Code on: BUILD (page 2-47).

P2PDEST

```
►►──┬──────────────────────────────────┬──────►◄
    └─P2PDEST=(──address──,─┬─D──┐──)──┘
                            ├─H──┤
                            └─DD─┘
```
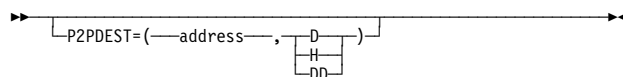
For Internet Protocol. Specifies the internet host address of the destination in a point-to-point connection.

Code on: IPLOCAL (page 2-157).

QUIET

Keyword	Description

P2PDEST

```
►►──┬──────────────────────────┬──►◄
    └─P2PDEST=(──address──,─┬─D──┬─)─┘
                           ├─H──┤
                           └─DD─┘
```
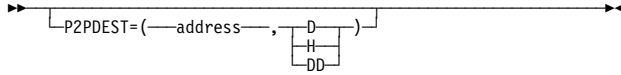
Frame-relay and channel internet connections, and internal NCP to Model 900 point-to-point IP connections. Specifies the destination internet address for the associated internet host defined by this PU definition statement.

Code on: PU (page 2-412)

QLIMIT

```
       ┌─QLIMIT=1────────┐
►►──┼──────────────────┼──►◄
       └─QLIMIT=requests─┘
```

Specifies the maximum number of callout requests NCP is to allow to accumulate on the queue for this dial set.

Code on: DIALSET (page 2-81).

QLOAD

```
       ┌─QLOAD=0────────┐
►►──┼─────────────────┼──►◄
       └─QLOAD=requests─┘
```

Specifies the number of callout requests NCP is to allow to accumulate on the queue for this dial set before using a line from the alternate dial set.

Code on: DIALSET (page 2-82).

| **QSIZE**
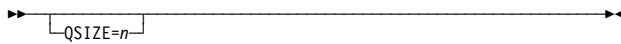
```
►►──┬──────────┬──►◄
    └─QSIZE=n──┘
```

For 3746 Model 900 frame-relay and ISDN congestion control. Specifies the 3746 Model 900 software queue threshold which, when it is reached, triggers the discarding of all of the last frames received regardless of how the Discard Eligible (DE) indicator is set.

Code on: GROUP, LINE (page 2-265).

| **QSIZEDE**

```
►►──┬────────────┬──►◄
    └─QSIZEDE=n──┘
```

For 3746 Model 900 frame-relay and ISDN congestion control. Specifies the 3746 Model 900 software queue threshold which, when it is reached, triggers the discarding of the last frames received with the Discard Eligible (DE) indicator set on.

Code on: GROUP, LINE (page 2-266).

QSZALERT

```
►►──┬─────────────────────────────────────────────────────────────────────┬──►◄
    │                        ┌─100-threshold_1%─┐   ┌─NONE──┐              │
    └─QSZALERT=(threshold_1%,─┼──────────────────┼─,─┼───────┼─)           │
                             └─threshold_2%──────┘   ├─NONE──┤
                                                     └─DEACT─┘
```
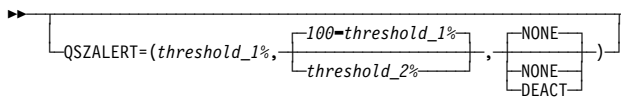
Specifies a buffer-utilization threshold, *threshold_1%*, at which NCP will alert the operator and determine whether a session or station is using an excessive number of buffers as defined by *threshold_2%*.

Code on: BUILD (page 2-47).

QUIET

```
       ┌─QUIET=NO──────┐
►►──┼────────────────┼──►◄
       └─QUIET=─┬─YES─┬─┘
               └─NO──┘
```
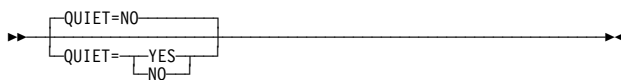
For start-stop lines in emulation mode. Specifies whether the control program is to use a long quiet time-out of 25.6 seconds when receiving from this line.

Code on: GROUP, LINE (page 2-265).

Keyword	Description

QUIETCT

```
      ┌─QUIETCT=0──────────┐
▶▶────┤                    ├────────────────◀◀
      └─QUIETCT=char_times─┘
```

For start-stop lines in emulation mode. Specifies, in character times, the interval NCP or EP is to wait between the end of a receive operation and the beginning of the following transmit operation.

Code on: GROUP (page 2-132).

RCVBUFC

```
▶▶────┬────────────────┬────────────────────◀◀
      └─RCVBUFC=bytes──┘
```

For NTRI physical lines. Specifies the NTRI buffer capacity for receiving data from a telecommunication link during one data transfer.

Code on: GROUP, LINE (page 2-266).

RCVCHAR

```
      ┌─RCVCHAR=4───┐
▶▶────┤             ├──────────────────────◀◀
      └─RCVCHAR=─┬─1─┬─┘
                 └─4─┘
```
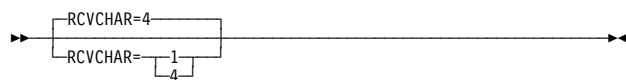
For start-stop lines in emulation mode. Specifies whether the line receives data in character mode or burst mode.

Code on: GROUP, LINE (page 2-267).

REDIAL

```
▶▶────┬──────────────────────────────────────────┬────◀◀
      │         ┌─0─┐   ┌─0─┐   ┌─0─┐             │
      └─REDIAL=(─┬──┬─,─┬──┬─,─┬──┬─,─┬──┬──)─────┘
                 └─m─┘  └─t1─┘ └─n─┘  └─t2─┘
```
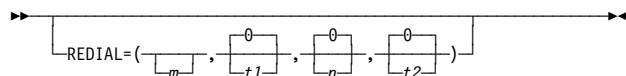
For network control mode. Specifies the number of dialing operations NCP is to perform in attempting to reach a device over a switched line and the pause desired between dialing operations.

Code on: GROUP, LINE (page 2-267).

REMLU

```
▶▶────┬──────────────────────────────────────────┬────◀◀
      └─REMLU=(name,─┬──────────┬─,─┬─────────┬─)─┘
                     └─modename─┘   └─SECONLY─┘
```
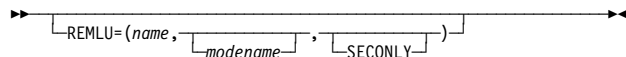
For Internet Protocol. Specifies the remote partner for this logical unit. When the SSCP-LU session for this NCST logical unit occurs, the value of REMLU is used to auto-initiate the LU-LU session.

Code on: LU (page 2-308).

REMOTTO
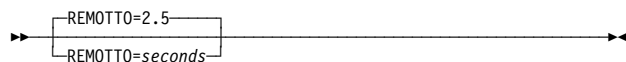
```
      ┌─REMOTTO=2.5─────┐
▶▶────┤                 ├──────────────────◀◀
      └─REMOTTO=seconds─┘
```

For Token-Ring logical lines. Specifies the Token-Ring reply timer (T1) value for stations connected to a remote Token Ring.

Code on: BUILD (page 2-48).

REMOTTO

```
▶▶────┬─────────────────┬──────────────────◀◀
      └─REMOTTO=seconds─┘
```

For Token-Ring logical subarea lines. Specifies the Token-Ring reply timer (T1) value for stations connected to a remote Token Ring.

Code on: GROUP, LINE (page 2-268), PU (page 2-413).

REMOVCOM

```
      ┌─REMOVCOM=NO─────┐
▶▶────┤                 ├──────────────────◀◀
      └─REMOVCOM=─┬─YES─┬─┘
                  └─NO──┘
```
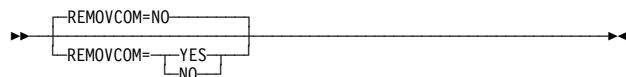
For the migration aid function. Specifies whether NDF is to remove comments added to the NEWDEFN file by the migration aid.

Code on: OPTIONS (page 2-347).

RETRYTO

Keyword	Description

REPLYTO

```
►►──┬─────────────────────────────┬──►◄
     └─REPLYTO=─┬─seconds─┬─────────┘
               └─NONE────┘
```
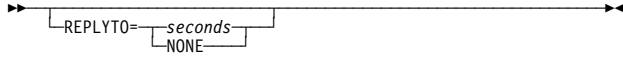
Specifies the reply time-out value for the lines in this line group.

Code on: GROUP (page 2-132).

RESERVE

```
       ┌─RESERVE=0───────┐
►►──┬──┴─────────────────┴──┬──────────────►◄
    └─RESERVE=lines──────────┘
```

Specifies how many lines in the dial set are to be reserved for incoming calls from devices.

Code on: DIALSET (page 2-82).

RESSCB

```
   ┌─RESSCB=0────────┐
►►─┴─────────────────┴──────────────────────►◄
   └─RESSCB=number───┘
```
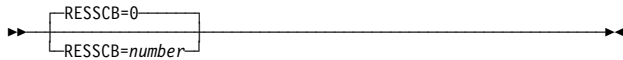
Specifies how many boundary session control blocks are to be reserved for this independent logical unit.

Code on: LU (page 2-309).

RETRIES

```
►►──┬───────────────────────────────────────────────────────┬──►◄
    │              ┌───┐      ┌─0─┐       ┌─6──┐              │
    └─RETRIES=─┬──(┴─m─┴──,───┴─t─┴──,─┬─n─┬──,┴─ml─┴──)─┬────┘
              └─NONE──────────────────────────────────────┘
```
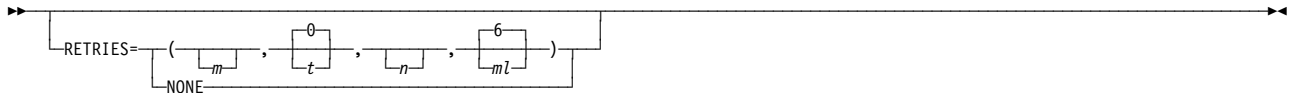
Specifies the number and grouping of attempts NCP is to make to recover from errors occurring during transmission over this link.

Code on: GROUP, LINE (page 2-269).

RETRIES
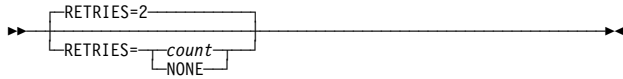
```
   ┌─RETRIES=2────────┐
►►─┴──────────────────┴──────────────────────►◄
   └─RETRIES=─┬─count─┬─┘
             └─NONE──┘
```

Specifies the maximum number of attempts NCP is to make to recover from text errors in message data sent to or received from an IBM 1050 terminal or an IBM 2740 terminal with record checking.

Code on: MTALCST (page 2-321).

RETRIES

```
►►──┬──────────────────────────────┬────────►◄
     └─RETRIES=(,─┬───┬──┬─────┬──)─┘
                 └─t─┘  └─,n─┘
```
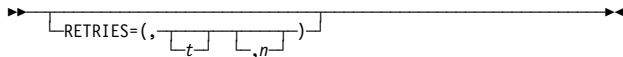
For a nonswitched link. Specifies, with RETRIES on the LINE definition statement, how many attempts NCP is to make to recover from errors occurring during transmission to or from this physical unit.

Code on: PU (page 2-414).

RETRIES
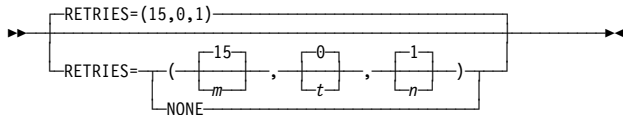
```
   ┌─RETRIES=(15,0,1)─────────────────────┐
►►─┴──────────────────────────────────────┴─►◄
   │          ┌─15─┐     ┌─0─┐     ┌─1─┐     │
   └─RETRIES=─┬─(─┴─m──┴──,─┴─t─┴──,─┴─n─┴──)─┘
             └─NONE─────────────────────────┘
```

Specifies how many attempts NCP is to make to recover from errors occurring during transmission over the link associated with this SDLCST definition statement.
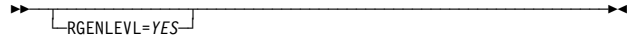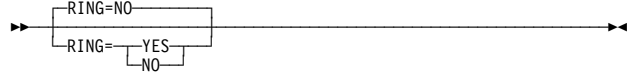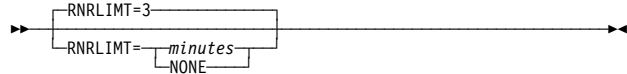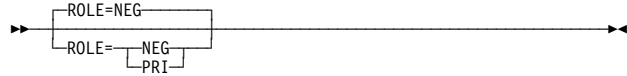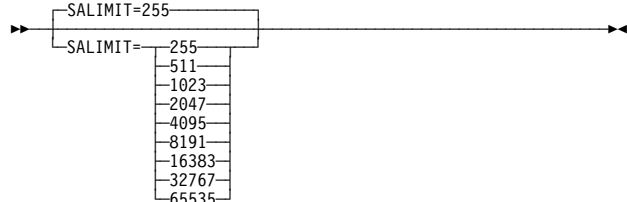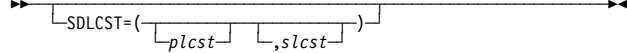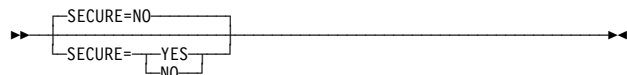
Code on: SDLCST (page 2-436).

RETRYTO

```
►►──┬──────────────────────┬────────────────►◄
     └─RETRYTO=seconds──────┘
```

For X.21 interface and network control mode. Specifies, in seconds, how long NCP is to wait after receiving a retry call progress signal before retrying a callout.

Code on: GROUP (page 2-133).

Keyword	Description

RGENLEVL

```
►►──┬────────────────┬──►◄
     └─RGENLEVL=YES─┘
```

Specifies that NDF is to replace the GENLEVEL keyword value with the default date and time of the NCP generation.

RING

```
         ┌─RING=NO──────┐
►►──┬─────────────────┬──►◄
     └─RING=─┬─YES─┬──┘
             └─NO──┘
```

For switched lines. Specifies whether NCP is to use the ring-indicator mode of an automatic answer operation.

Code on: GROUP, LINE (page 2-272).

RNRLIMT

```
         ┌─RNRLIMT=3──────────┐
►►──┬──────────────────────┬──►◄
     └─RNRLIMT=─┬─minutes─┬─┘
                └─NONE────┘
```

For SDLC lines; Token-Ring, frame-relay, and ISDN logical lines. Specifies how long NCP is to allow a station to refuse to accept data before identifying the station as inoperative.

Code on: GROUP (page 2-134).

ROLE

```
         ┌─ROLE=NEG──────┐
►►──┬─────────────────┬──►◄
     └─ROLE=─┬─NEG─┬──┘
             └─PRI─┘
```

For switched SDLC lines. Specifies the data link control role that the station is to assume at CONTACT time if format-3 XIDs are exchanged during the connect sequence.

Code on: GROUP, LINE, PU (page 2-414).

SALIMIT

```
         ┌─SALIMIT=255──────┐
►►──┬───────────────────┬──►◄
     └─SALIMIT=─┬─255───┬─┘
               ├─511───┤
               ├─1023──┤
               ├─2047──┤
               ├─4095──┤
               ├─8191──┤
               ├─16383─┤
               ├─32767─┤
               └─65535─┘
```

Specifies the largest subarea address supported by this node in the native or non-native network.

Code on: BUILD (page 2-48), NETWORK (page 2-337).

SAVEADDR

```
►►──┬──────────────────┬──►◄
     └─SAVEADDR=─┬─NO──┬─┘
                 └─YES─┘
```

For the migration aid function. Specifies whether hardware-specific addresses coded for ADDRESS, AUTO, DUALCOM, DYNADMP, HICHAN, and LOCHAN on the BUILD and LINE definition statements in the input generation definition are to be preserved in the output generation definition.

Code on: OPTIONS (page 2-347).

SDLCST

```
►►──┬─────────────────────────────┬──►◄
     └─SDLCST=(─┬───────┬─┬────────┬─)─┘
               └─plcst─┘ └─,slcst─┘
```

For point-to-point subarea lines. Specifies which SDLC selection table entry NCP is to use when it operates in a primary (*plcst*) or secondary (*slcst*) state.

Code on: GROUP, LINE (page 2-273).

SECURE

```
         ┌─SECURE=NO──────┐
►►──┬──────────────────┬──►◄
     └─SECURE=─┬─YES─┬──┘
              └─NO──┘
```

For switched, duplex, start-stop lines in network control mode. Specifies whether NCP is to use the secure option for this line.

Code on: GROUP, LINE (page 2-273).

SESSION

Keyword	Description

SEGMENT

```
►►──┬──────────────────┬──►◄
    └─SEGMENT=─┬─1─┬───┘
              └─2─┘
```

Specifies which link segment this device is on. This allows NCP to set up LPDA2 tests.

Code on: CLUSTER (page 2-74), PU (page 2-415), TERMINAL (page 2-469).

SERVLIM

```
►►──┬────────────────────┬──►◄
    └─SERVLIM=entries────┘
```

For nonswitched multipoint lines and SDLC links in network control mode. Specifies the maximum number of entries in the SOT NCP is to check each time NCP performs service seeking for start-stop or BSC lines. Also specifies the maximum number of regular SOT scans that NCP is to make for normal servicing of physical and logical units on this line before making a special scan of the table for SDLC lines.

Code on: GROUP, LINE (page 2-274).

SERVLIM

```
►►──┬─────────────────────────────────────┬──►◄
    └─SERVLIM=(─┬─────────┬─┬──────────┬─)┘
               └─entries─┘ └─,seconds─┘
```
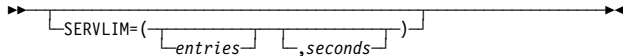
For a primary SDLC link. Specifies the maximum number of regular scans NCP is to make of the SOT for normal servicing of the physical unit on the link before making a special scan for outstanding host commands or specifies the minimum time between special scans of the SOT.

Code on: SDLCST (page 2-437).

SERVPRI

```
        ┌─SERVPRI=OLD───────┐
►►──────┤                   ├──►◄
        └─SERVPRI=─┬─OLD─┬──┘
                   └─NEW─┘
```

For multipoint start-stop and BSC lines in network control mode. Specifies whether NCP is to give priority to servicing current sessions or to establishing new ones on this line.

Code on: GROUP, LINE (page 2-275).

SESSACC

```
        ┌─SESSACC=NO──────────────────────────────────────────────────────────────────────────────┐
►►──────┤                                                                                          ├──►◄
        │                              ┌─2147483646─┐ ┌─32767─┐          ┌────,────┐                │
        └─SESSACC=─┬─(YES,─┬─────┬─,─┬─────┬─,─┬──────────┬─,─┬───────┬─,─┬───────┬─,─▼─┬─────────┬─)┘
                   │       ├─PLU─┤   └─nscnum─┘ └─bytenum─┘   └─piunum─┘  └─nscx──┘   └─piudist─┘
                   │       ├─SLU─┤
                   │       ├─ALL─┤
                   │       └─DEFER─┘
                   └─NO────┘
```
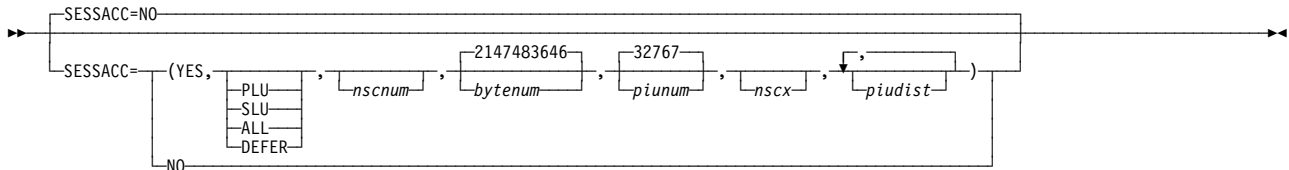
Specifies whether NCP is to session accounting and, if so, which session accounting characteristics are to be used.
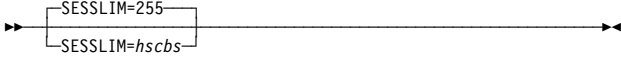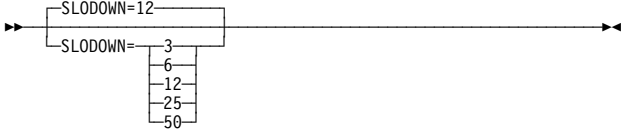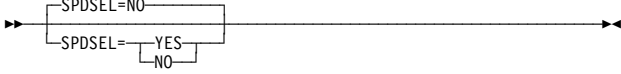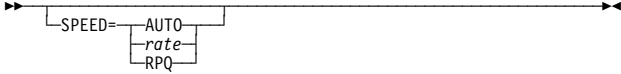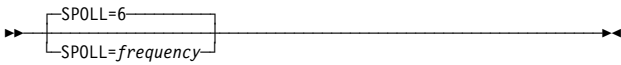
Code on: BUILD (page 2-49).

SESSION

```
        ┌─SESSION=1──────────┐
►►──────┤                    ├──►◄
        └─SESSION=sessions───┘
```

For nonswitched, multipoint start-stop, and BSC lines in network control mode. Specifies the number of sessions NCP will attempt to maintain concurrently on this line.

Code on: GROUP, LINE (page 2-275).

Keyword	Description

SESSLIM

```
      ┌─SESSLIM=255──┐
►►────┤              ├────────────────────────►◄
      └─SESSLIM=hscbs┘
```

For a gateway NCP. Specifies how many HSCBs can be assigned to any address in the native or non-native network.

Code on: BUILD (page 2-51), NETWORK (page 2-337).

SESSREQ

```
      ┌─SESSREQ=NO───────┐
►►────┤                  ├────────────────────►◄
      └─SESSREQ=─┬─YES─┬─┘
                 └─NO──┘
```

For an IBM 3270 on a BSC line. Specifies whether a session with this terminal must exist before NCP can start polling for data.

Code on: TERMINAL (page 2-469).

SLODOWN

```
      ┌─SLODOWN=12──────┐
►►────┤                 ├─────────────────────►◄
      └─SLODOWN=─┬─3──┐─┘
                 ├─6──┤
                 ├─12─┤
                 ├─25─┤
                 └─50─┘
```

Specifies the NCP slowdown threshold as a percentage of the total number of NCP buffers.

Code on: BUILD (page 2-51).

SNETMASK

```
►►──┬──────────────────────────┬──────────────►◄
    └─SNETMASK=address_mask─────┘
```

For Internet Protocol. Defines the subnetwork mask that is used to route IP datagrams to subnetworks.

Code on: IPLOCAL (page 2-158), IPROUTE (page 2-169), PU (page 2-415).

SPDSEL

```
      ┌─SPDSEL=NO───────┐
►►────┤                 ├─────────────────────►◄
      └─SPDSEL=─┬─YES─┬─┘
                └─NO──┘
```

For start-stop and BSC lines in network control mode. Specifies whether the data rate of a dual-rate modem can be changed by a request from the access method.

Code on: GROUP, LINE (page 2-276).

SPEED

```
►►──┬────────────────────┬─────────────────────►◄
    └─SPEED=─┬─AUTO─┬─────┘
             ├─rate─┤
             └─RPQ──┘
```

Specifies the data rate for this line in bits per second.

Code on: GROUP, LINE (page 2-276).

SPEED

```
►►──┬─────────────┬────────────────────────────►◄
    └─SPEED=rate──┘
```

Specifies the data rate in bits per second for the MTA lines associated with the entry defined.

Code on: MTALCST (page 2-322).

SPOLL

```
      ┌─SPOLL=6───────────┐
►►────┤                   ├───────────────────►◄
      └─SPOLL=frequency───┘
```

For frame-relay link station subports and ISDN physical units. Specifies the minimum frequency at which LMI Status Enquiry Full messages are to be sent on the frame-relay physical line to the adjacent node.

Code on: GROUP, LINE, PU (page 2-416).

SPSHIFT

```
      ┌─SPSHIFT=NO──────┐
►►────┤                 ├─────────────────────►◄
      └─SPSHIFT=─┬─YES─┬─┘
                 └─NO──┘
```

For teletypewriter lines in network control mode. Specifies whether NCP interprets space characters received from 83B3, WU 115A, or WTTY terminals as downshift characters.

Code on: GROUP, LINE (page 2-279).

SUBPORTS

Keyword	Description

SRCHI
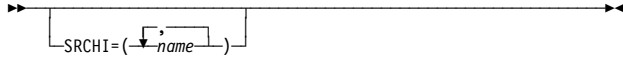
```
►►─┬──────────────────────────┬─►◄
   └─SRCHI=(──┬─,─┬──name──)─┘
             └───┘
```

Identifies source code for user-defined control blocks and the tables assembled with NCP control blocks and tables. This source code may reside anywhere in communication controller storage.

Code on: GENEND (page 2-93).

SRCLO
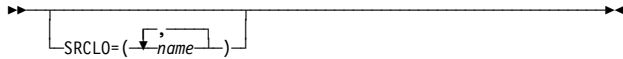
```
►►─┬──────────────────────────┬─►◄
   └─SRCLO=(──┬─,─┬──name──)─┘
             └───┘
```

Identifies source code for user-defined control blocks and the tables assembled with NCP control blocks and tables. This source code will reside in the lower 4MB of communication controller storage.

Code on: GENEND (page 2-93).

SRETRY

```
►►─┬──SRETRY=15──────┬─►◄
   └──SRETRY=retries─┘
```

Specifies the maximum number of retries to be attempted by secondary physical units on this line.

Code on: GROUP, LINE (page 2-280).

SRT

```
►►─┬──SRT=(32768,32768,YES)──────────────────┬─►◄
   │                                          │
   └─SRT=(─┬─32768─┬─,─┬─32768─┬─,─┬─YES─┬─)──┘
           └──m────┘   └──n────┘   ├─YES─┤
                                   └─NO──┘
```
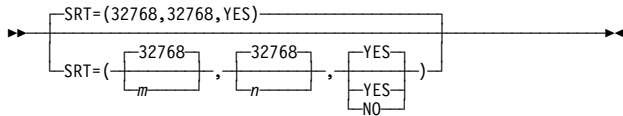
For SDLC lines. Specifies a threshold value for the total number of transmissions and the total number of error retries associated with the physical unit.

Code on: GROUP, LINE, PU (page 2-416).

SRT

```
►►─┬──SRT=(65535,255)──────────────┬─►◄
   │                                │
   └─SRT=(─┬─65535─┬─,─┬─255─┬─)────┘
           └──m────┘   └──n──┘
```
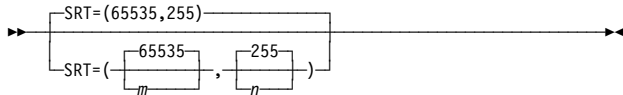
Specifies a threshold value for the total number of retries associated with a terminal.

Code on: GROUP, LINE, TERMINAL (page 2-470), COMP.

SSAP

```
►►─┬──SSAP=(4,H)──────────────────┬─►◄
   │                              │
   └─SSAP=(─┬─4────┬─,─┬─H─┬─)────┘
           └─ssap─┘   ├─D─┤
                      └─H─┘
```
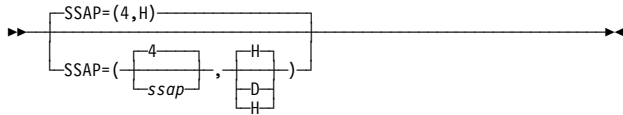
For Token-Ring logical subarea PUs. Specifies the source service access point (SSAP) for a Token-Ring logical subarea connection.

Code on: PU (page 2-418).

SUBAREA

```
►►──SUBAREA=address──────────────────►◄
```

Required. Specifies the subarea address assigned to NCP or PEP.

Code on: BUILD (page 2-52).

SUBAREA

```
►►──SUBAREA=address──────────────────►◄
```

Required when you code the NETWORK definition statement. Specifies the subarea address of the gateway NCP as it is known in the non-native network.
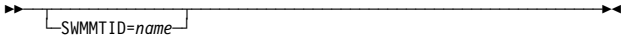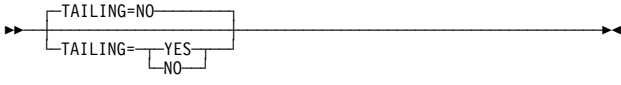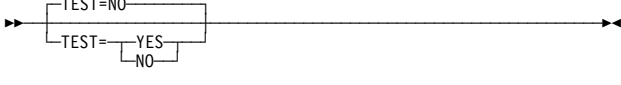
Code on: NETWORK (page 2-337).

SUBPORTS

```
►►──SUBPORTS=(primary1,primary2,─┬─────────┬─,─┬─────────┬─)──►◄
                                 └─backup1─┘   └─backup2─┘
```
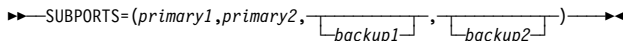
Required when you code the FRSESET definition statement. Defines a set of primary frame-relay frame handler subports and optional substitute frame-relay frame handler subports.

Code on: FRSESET (page 2-91).

Keyword	Description

SWMMTID

```
►►──┬──────────────────┬──►◄
     └─SWMMTID=name─────┘
```

For switched subarea lines. Specifies the name of a switched SMMF table defined by a group of SWMMT definition statements.

Code on: GROUP, LINE (page 2-280).

TADDR

```
►►──┬──────────────────┬──►◄
     └─TADDR=address────┘
```

Specifies a symbolic link station address for this NCP when it is a type 4 physical unit on a subarea link.

Code on: GROUP, LINE (page 2-280).

TADDR

```
►►──┬──────────────────┬──►◄
     └─TADDR=chars──────┘
```

For secondary SDLC subarea links. Specifies the address of the secondary communication controller where NCP resides, as a 2-digit hexadecimal representation of a single EBCDIC SDLC device address

Code on: SDLCST (page 2-438).

TAILING

```
►►──┬─TAILING=NO────────┬──►◄
     └─TAILING=─┬─YES─┬──┘
                └─NO──┘
```

Specifies whether this line attaches to a second-level LPDA circuit.

Code on: GROUP, LINE (page 2-282).

TERM

```
►►──TERM=type──────────────►◄
```

Specifies the type of device NCP or EP will communicate with over the line.

Code on: GROUP, LINE (page 2-282), CLUSTER, TERMINAL (page 2-470).

TEST

```
►►──┬─TEST=NO───────────┬──►◄
     └─TEST=─┬─YES─┬─────┘
             └─NO──┘
```

For PEP. Specifies whether PEP is to include the panel-initiated line test.

Code on: BUILD (page 2-52).

TEXTTO

```
►►──┬──────────────────────┬──►◄
     └─TEXTTO=─┬─seconds─┬──┘
               └─NONE────┘
```

Specifies a text time-out value for the lines in a group.

Code on: GROUP (page 2-134).

TGBXTRA

```
►►──┬──────────────────┬──►◄
     └─TGBXTRA=count────┘
```

For dynamic path update. Specifies how many extra transmission group control blocks are to be generated in addition to those generated by the PATH definition statements for the native network.

Code on: BUILD (page 2-52).

TGBXTRA

```
►►──┬──────────────────┬──►◄
     └─TGBXTRA=tgbs─────┘
```

Specifies how many extra transmission group control blocks NDF is to generate after processing all PATH definition statements for this non-native network.

Code on: NETWORK (page 2-338).

TITIMER

Keyword	Description

TGCONF

```
        ┌TGCONF=(MULTI,NORMAL)──────────────────┐
►►──────┤                                       ├──►◄
        └─TGCONF=(─┬───────┬─,─┬─NORMAL──┬─)─────┘
                   ├─MULTI─┤   └─STANDBY,┘
                   └─SINGLE┘       ┌─nn──┐
                                   └─NONE─┘
```
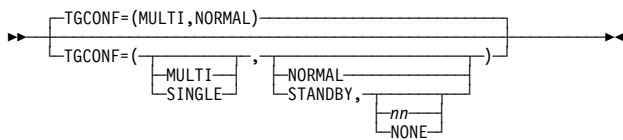
Specifies whether this transmission group can include multiple SDLC, Token-Ring, or frame-relay lines.

Code on: GROUP, LINE, PU (page 2-418).

TGN

```
        ┌─TGN=1─────────┐
►►──────┤               ├──►◄
        └─TGN=─┬─number─┬┘
               └─ANY────┘
```

For type 4 or 5 physical units. Specifies the transmission group number (TGN) of the subarea link associated with this physical unit, or specifies that the TGN is to be assigned during the contact procedure with an adjacent subarea.

Code on: GROUP, LINE, PU (page 2-419).

TIME

```
        ┌─TIME=YES───┐
►►──────┤            ├──►◄
        └─TIME=─┬─YES┬┘
               └─NO─┘
```
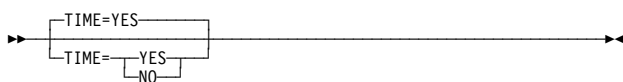
Specifies whether NCP is to insert the current time in a block of data.

Code on: DATETIME (page 2-78).

TIMEOUT
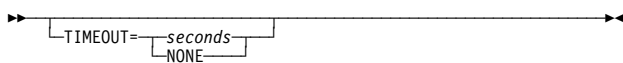
```
        ┌──────────────────────┐
►►──────┴─TIMEOUT=─┬─seconds─┬─┴──►◄
                   └─NONE────┘
```

For channel links. Specifies how long NCP is to wait for a response to an attention signal it sent to the host before initiating channel discontact, or that NCP is to wait an unlimited time for a response from the host.

Code on: GROUP, LINE (page 2-284), PU (page 2-420).

TIMER

```
►►──────┴─TIMER=(error,─┬─────┬─,stap,lstap)─┴──►◄
                        └─ras─┘
```
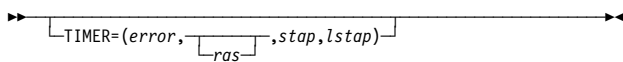
For user-written line control and NTRI physical lines. Specifies the entry points of user-written or NTRI timer service routines.

Code on: GROUP (page 2-135).

TIMERS

```
        ┌─TIMERS=(10,15)────────────────────┐
►►──────┤                                   ├──►◄
        └─TIMERS=(─┬─10───┬─,─┬─15───┬─)─────┘
                   └─t391─┘   └─t392─┘
```
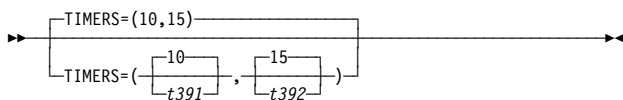
For frame-relay link station subports, ISDN physical units, and NTRI frame handler logical link station subports. Specifies the length of timer intervals used by the LMI protocol.

Code on: GROUP, LINE, PU (page 2-421).

TITIMER

```
        ┌─TITIMER=60──────────────┐
►►──────┤                         ├──►◄
        └─TITIMER=time_interval───┘
```

For Token-Ring, frame-relay, and ISDN physical lines. Specifies the inactivity timer (TI) interval, in seconds, that NCP is to wait for a transmitted frame from an adjacent station before sending a poll command.

Code on: LINE (page 2-285).

Keyword	Description

TMODEL

```
►►──┬──────────────────────────────┬──►◄
    └─TMODEL=─┬─3745-130─┬─────────┘
             ├─3745-150─┤
             ├─3745-160─┤
             ├─3745-170─┤
             ├─3745-17A─┤
             ├─3745-210─┤
             ├─3745-21A─┤
             ├─3745-310─┤
             ├─3745-31A─┤
             ├─3745-410─┤
             ├─3745-41A─┤
             ├─3745-610─┤
             └─3745-61A─┘
```
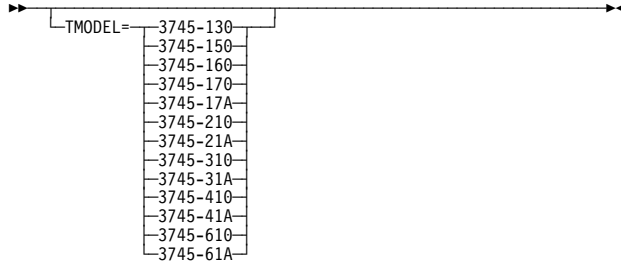
Required for the migration aid function. Specifies the controller model to which you are migrating.

Code on: OPTIONS (page 2-348).

TMRTICK

```
►►──┬────────────────────────────┬──►◄
    └─TMRTICK=(─┬──────┬─)────────┘
               └─┬──name─┘
                 └─,─┘
```

For user-written code, NTRI, or 3745 frame relay. Specifies the entry point of up to five timer-tick service routines.

Code on: GENEND (page 2-95).

TO

```
►►──TO=name──────────────────────────►◄
```

Required when you code the ADD definition statement. Specifies the name of the link defined in the generation definition to which a new physical unit will be dynamically added, or the name of the physical unit defined in the generation definition to which a new logical unit will be dynamically added.

Code on: ADD (page 2-3).

TOCONV

```
►►──┬────────────────────┬───────────►◄
    └─TOCONV=─┬─YES─┬─────┘
             └─NO──┘
```
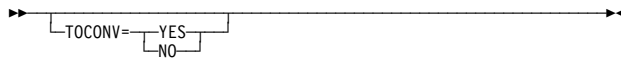
Replaces the LOCALTO and REMOTTO keywords on Token-Ring and frame relay lines with T1TIMER keyword.

Code on: OPTIONS (page 2-349).

TRACE

```
►►──┬─TRACE=NO──────────────────────┬─►◄
    └─TRACE=─┬─(YES,─┬─50──────┬─)─┬─┘
            │       └─entries─┘   │
            └─NO──────────────────┘
```
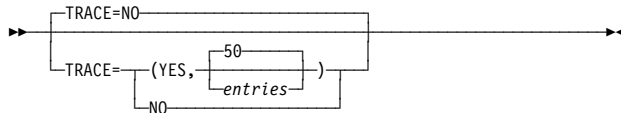
Specifies whether NCP is to include the address trace.

Code on: BUILD (page 2-53).

| **TRANSBUF**
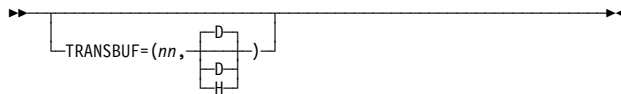
```
►►──┬────────────────────────────┬───►◄
    └─TRANSBUF=(nn,─┬─D─┬─)───────┘
                   ├─D─┤
                   └─H─┘
```

Specifies the number of frames queued for a TSS or HPTSS adapter awaiting transmission over a 3745 frame-relay physical line.

Code on: LINE (page 2-274).

TRANSFR

```
►►──┬──────────────────┬─────────────►◄
    └─TRANSFR=buffers───┘
```

For network control mode. Specifies the default value for TRANSFR on the LINE definition statement.

Code on: BUILD (page 2-53).

TRANSFR

```
►►──┬──────────────────┬─────────────►◄
    └─TRANSFR=buffers───┘
```

For ESCON physical link stations. Specifies the number of NCP buffers corresponding to the maximum amount of data NCP can receive from the station during a single data transfer operation.

Code on: GROUP, LINE (page 2-286), PU (page 2-422).

TTDCNT

Keyword	Description

TRANSFR

```
►►─┬──────────────────┬─►◄
   └─TRANSFR=buffers──┘
```

Specifies the maximum number of buffers NCP is to use to receive message text from a terminal before transferring filled buffers to the host processor.

Code on: MTALCST (page 2-322).

TRCPIU

```
►►─┬─TRCPIU=2000───┬─►◄
   └─TRCPIU=bytes──┘
```

For network control mode. Specifies the size of the line trace and scanner interface trace PIUs.

Code on: BUILD (page 2-54).

TRDATA
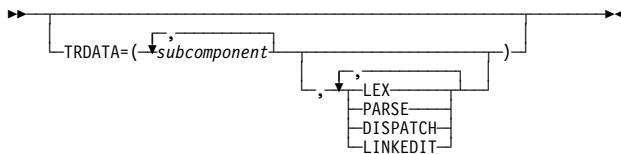
```
►►─┬────────────────────────────────────────────┬─►◄
   │                    ,                        │
   └─TRDATA=(─▼─subcomponent─┬──────────────┬──)─┘
                             │      ,       │
                             └─▼─LEX──────┬─┘
                               ├─PARSE────┤
                               ├─DISPATCH─┤
                               └─LINKEDIT─┘
```

Specifies the subcomponent or data areas for which the data trace is to be activated.

Code on: OPTIONS (page 2-349).

TRGLOB

```
►►─┬───────────────────────────────┬─►◄
   │             ,                  │
   └─TRGLOB=─┬(─▼─global_trace──)─┬─┘
            ├─HALT──────────────┤
            └─RESUME────────────┘
```

Specifies the global variables for which the global trace is to be activated.

Code on: OPTIONS (page 2-350).

TRPARM

```
►►─┬──────────────────────────────┬─►◄
   │              ,                │
   └─TRPARM=(─▼─subcomponent──┬──)─┘
                     └─,SK─┘
```
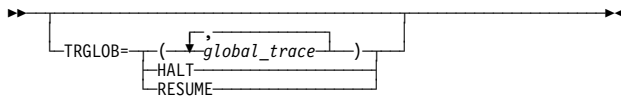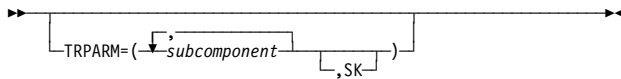
Specifies the subcomponent or definition statement and keyword routines for which the I/O parameter trace is to be activated.

Code on: OPTIONS (page 2-350).

TRPROC

```
►►─┬──────────────────────────────┬─►◄
   │              ,                │
   └─TRPROC=(─▼─subcomponent──┬──)─┘
                     └─,SK─┘
```

Specifies the subcomponent or definition statement and keyword routines for which the procedure entry and exit trace is to be activated.

Code on: OPTIONS (page 2-350).

TRSNAP

```
►►─┬────────────────────────┬─►◄
   │          ,             │
   └─TRSNAP=(─▼─KVT──────┬─)─┘
             ├─SYMBOLS──┤
             └─DATABASE─┘
```
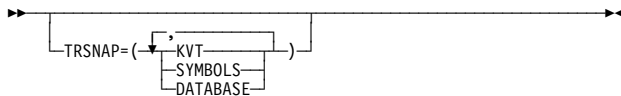
Specifies which major data structures NDF is to print.

Code on: OPTIONS (page 2-351).

TRSPEED

```
►►─┬─TRSPEED=─┬─4──┬─┬─►◄
              └─16─┘
```
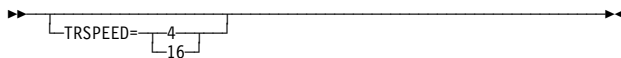
Specifies, in Mbps, the data speed on the Token Ring attached to this line.

Code on: GROUP, LINE (page 2-288).

TTDCNT
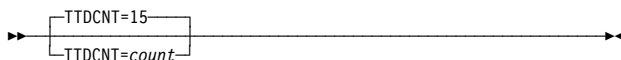
```
►►─┬─TTDCNT=15────┬─►◄
   └─TTDCNT=count─┘
```

For BSC lines in network control mode. Specifies how many times NCP is to accept the BSC temporary text-delay sequence from a device before aborting the operation.

Code on: GROUP (page 2-136).

| Keyword | Description |

TUSGTIER

```
►►─TUSGTIER=──┬─1──┬────────────────────────┬──►◄
              │    ├─ -S ──(1)              │
              │    └─ -TS                   │
              │         ┌─ -TB ─(2)         │
              ├─2────┬──┴──────────────┤    │
              ├─2.5──┼─ -S ─────────────┤   │
              ├─3────┼─ -TB ────────────┤   │
              ├─4────┼─ -TD ────────────┤   │
              └─5────┴─ -TS ────────────┘   │
```
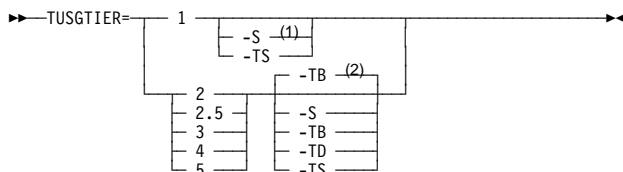
For the migration aid function. Specifies the NCP usage tier to which you are migrating.

Code on: OPTIONS (page 2-351).

[1] For usage tier 1, the -S suffix is not valid for 3745-410/41A/610/61A.

[2] Defaults for usage tiers 2 through 5:

- -TB is the default for tiers 2 and 2.5, but only if MODEL=3745-41A or 3745-61A. Otherwise there is no default for tiers 2 and 2.5.

- -TB is the default for tier 3.

- -TB is not a default for usage tiers 4 and 5. There is no default for usage tiers 4 and 5.

TVERSION

```
►►─TVERSION=──┬─V7R3──┬──►◄
              └─V7R3F─┘
```

For the migration aid function. Specifies the NCP version and release to which you are migrating.

Code on: OPTIONS (page 2-352).

TWXID

```
      ┌─TWXID=(XON,XOFF)─────────────────┐
►►────┤                                  ├──►◄
      │         ┌─XON──┐      ┌─XOFF───┐  │
      └─TWXID=(─┼─XON──┼──,───┼─XOFF───┼─)┘
                └inchars┘      └outchars┘
```
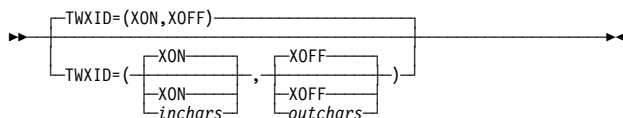
For TWX terminals on switched lines in network control mode. Specifies the characters used for the identifier answerback sequence (or identifier) and the prompting sequence (or PROMPT) sent as part of each TWX terminal READ operation.

Code on: BUILD (page 2-54).

TYPE

```
►►──┬──────────────┬──►◄
    └─TYPE=─┬─NCP─┐ 
            ├─PEP─┤
            └─EP──┘
```

Specifies, for the lines in this line group, whether (1) all operate in network control mode, (2) all operate in emulation mode, (3) some operate in one mode and some in another, or (4) some operate in either mode.

Code on: GROUP (page 2-136).

TYPE

```
►►──┬──────────────┬──►◄
    └─TYPE=─┬─NCP─┐ 
            ├─PEP─┤
            └─EP──┘
```
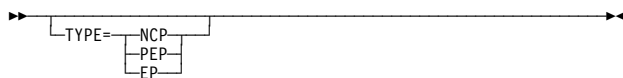
Specifies whether this line operates in network control mode, emulation mode, or both.

Code on: LINE (page 2-288).

TYPE

```
      ┌─TYPE=SSCP───┐
►►────┤             ├──►◄
      └─TYPE=─┬─SSCP──┐
              └─NCPLU─┘
```
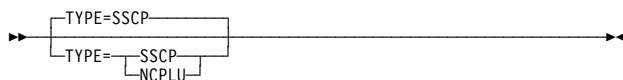
Specifies whether this NAU functions as an SSCP or as an NCP logical unit.

Code on: NCPNAU (page 2-330).

TYPGEN

```
►►─TYPGEN=──┬─NCP───┐──►◄
            ├─NCP-R─┤
            └─PEP───┘
```

Required. Specifies whether the communication controller is channel-attached (NCP) or not (NCP-R), and whether EP functions are to be included (PEP).

Code on: BUILD (page 2-55).

Keyword	Description

TYPSYS
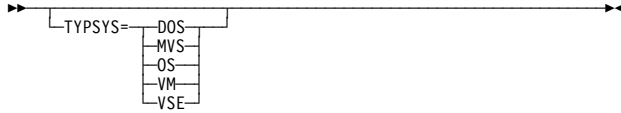
```
►►─┬─────────────────────────┬─►◄
   └─TYPSYS=─┬─DOS─┬─────────┘
            ├─MVS─┤
            ├─OS──┤
            ├─VM──┤
            └─VSE─┘
```

Specifies whether this NCP will be generated under an MVS (TYPSYS=MVS or TYPSYS=OS), VM (TYPSYS=VM), or VSE (TYPSYS=VSE or TYPSYS=DOS) operating system.

Code on: BUILD (page 2-55).

T1TIMER

```
►►─┬──────────────────────────────────────┬─►◄
   └─T1TIMER=(─┬─────────┬─,─┬─2.5─────┬─)─┘
              └─localt1─┘   └─remott1─┘
```
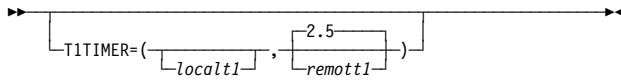
For Token-Ring, frame-relay, and ISDN logical resources. Specifies, in seconds, the reply timer (T1) values for stations connected to Token-Ring or frame-relay networks.

Code on: BUILD (page 2-56), LINE (page 2-289), PU (page 2-423).

T2TIMER
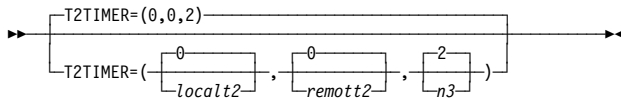
```
►►─┬─T2TIMER=(0,0,2)──────────────────────────────────┬─►◄
   └─T2TIMER=(─┬─0───────┬─,─┬─0───────┬─,─┬─2──┬─)─┘
              └─localt2─┘   └─remott2─┘   └─n3─┘
```

For Token-Ring logical lines. Specifies the Token-Ring acknowledgment timer (T2) values and the maximum number of information frames NCP can receive without sending an acknowledgment.

Code on: BUILD (page 2-56).

T2TIMER
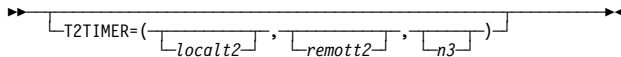
```
►►─┬────────────────────────────────────────────────┬─►◄
   └─T2TIMER=(─┬─────────┬─,─┬─────────┬─,─┬────┬─)─┘
              └─localt2─┘   └─remott2─┘   └─n3─┘
```

For Token-Ring, frame-relay, and ISDN logical lines. Specifies the Token-Ring acknowledgment timer (T2) values and the maximum number of I-frames NCP can receive without sending an acknowledgment.

Code on: GROUP, LINE (page 2-291), PU (page 2-424).

UACB

```
►►─┬─────────────────────────────┬─►◄
   └─UACB=(name1─┬─────────┬─)─┘
                └─,name2─┘
```
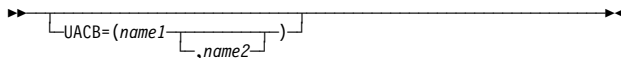
For user-written line control, NTRI, and frame relay. Specifies the names of the user, NTRI, or frame-relay adapter control blocks associated with this line.

Code on: LINE (page 2-292).

UACCTNG

```
►►─┬───────────────────────────┬─►◄
   │         ┌─,──────┐         │
   └─UACCTNG=(─▼─name─┴─)─┘
```
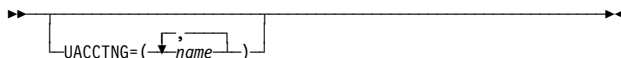
Specifies the entry points of accounting notification routines for IBM special products or user-written code.

Code on: GENEND (page 2-95).

UCCB

```
►►─┬──────────────┬─►◄
   └─UCCB=name─┘
```

For user-written code. Specifies the name of the user channel control block associated with this logical unit for channel operations performed by user-written code.
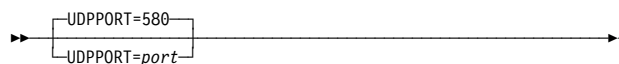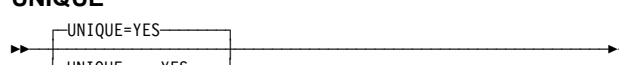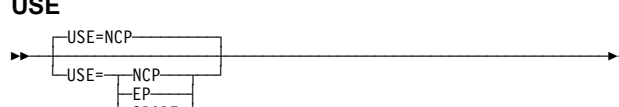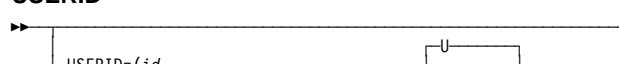
Code on: LU (page 2-309).

UCHAN

```
►►─┬─UCHAN=NO──────────┬─►◄
   └─UCHAN=─┬─YES─┬─┘
           └─NO──┘
```
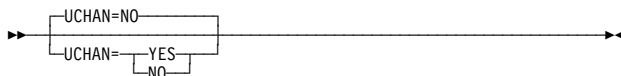
Specifies whether user-written code is provided for channel adapter logical addresses in the communication controller.

Code on: BUILD (page 2-58).

Keyword	Description

UDPPORT

```
     ┌─UDPPORT=580─┐
►►───┤             ├───►◄
     └─UDPPORT=port─┘
```

Specifies the User Datagram Protocol (UDP) port number used for communication between the NCP internet router and the NCPROUTE server of the IBM TCP/IP host.

Code on: IPOWNER (page 2-164).

UGLOBAL

```
                    ┌─,──┐
►►──┬──────────────────────┬──►◄
    └─UGLOBAL=(─▼─name─┴─)─┘
```

Specifies the entry points of IBM special products or user-written code, NTRI, or 3745 frame-relay routines that act as global routers.

Code on: GENEND (page 2-95).

UNIQUE

```
     ┌─UNIQUE=YES─┐
►►───┤            ├───►◄
     └─UNIQUE=─┬─YES─┤
               └─NO──┘
```

Specifies whether the 12-character medium access control (MAC) address, which is defined for the ADDR keyword and corresponds to the physical token-ring connection in the adjacent node, is a unique locally administered address.

Code on: PU (page 2-426).

UNITXC

```
     ┌─UNITXC=YES─┐
►►───┤            ├───►◄
     └─UNITXC=─┬─YES─┤
               └─NO──┘
```

For start-stop and BSC lines in emulation mode. Specifies whether NCP or EP is to signal a unit-exception status to the host when the an EOT is received from this line.

Code on: GROUP, LINE (page 2-293).

USE

```
     ┌─USE=NCP──┐
►►───┤          ├───►◄
     └─USE=─┬─NCP──┤
            ├─EP───┤
            ├─SPARE┤
            └─REDEF┘
```

For TYPE=PEP, specifies whether this line is to operate initially in network control mode (USE=NCP) or emulation mode (USE=EP). For TYPE=NCP, specifies whether this line is initially a spare line (USE=SPARE), a redefinable line (USE=REDEF), or neither (USE=NCP).

Code on: GROUP, LINE (page 2-293).

USERGEN

```
                    ┌─,──┐
►►──┬──────────────────────┬──►◄
    └─USERGEN=(─▼─name─┴─)─┘
```

Specifies the names of the generation application load modules that use the NDF standard attachment facility.

Code on: OPTIONS (page 2-352).

USERID

```
►►──┬──────────────────────────────────────────────────────────┬──►◄
    │                                        ┌─U──────┐         │
    └─USERID=(id,─┬───────┬─,─┬─────────┬─,─┤         ├─,─┬──────┬─)─┘
                  └─entry─┘   └─NORECMS─┘   └─traceid─┘   └─entry2─┘
```

For user-written or NTRI code. Specifies an identification symbol given to user-defined or NTRI control blocks associated with the line group.

Code on: GROUP (page 2-137).

VPACING

Keyword	Description

USGTIER

```
       ┌─ 1 ─┬──────────────┐
►►─USGTIER=─┤     ├─ -S ─(1)─┤ ─►◄
       │     └─ -TS ─────────┤
       │        ┌─ -TB ─(2)─┐│
       ├─ 2 ───┼─ -S ───────┤
       ├─ 2.5 ─┼─ -TB ──────┤
       ├─ 3 ───┼─ -TD ──────┤
       ├─ 4 ───┴─ -TS ──────┤
       └─ 5 ────────────────┘
```
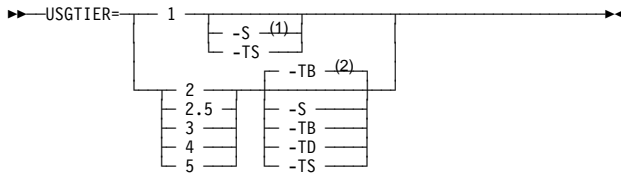
Required. Specifies the usage tier for this NCP. Not valid for EP standalone.

Code on: BUILD (page 2-59).

[1] For usage tier 1, the -S suffix is not valid for 3745-410/41A/610/61A.

[2] Defaults for usage tiers 2 through 5:

- -TB is the default for tiers 2 and 2.5, but only if MODEL=3745-41A or 3745-61A. Otherwise there is no default for tiers 2 and 2.5.

- -TB is the default for tier 3.

- -TB is not a default for usage tiers 4 and 5. There is no default for usage tiers 4 and 5.

VERSION

```
       ┌─ VxRy ──┐
►►─VERSION=─┤         ├─────────────►◄
       └─ VxRyF ─┘
```

Required. Specifies the NCP or EP version and release to be generated.

Code on: BUILD (page 2-61).

VIROWNER

```
►►─┬─────────────────┬──────────────────►◄
   └─ VIROWNER=name ─┘
```

Specifies the name of the user-written generation load module responsible for processing the resources defined by this GROUP definition statement.

Code on: GROUP (page 2-138).

VIROWNER

```
►►─┬─────────────────┬──────────────────►◄
   └─ VIROWNER=name ─┘
```

Specifies the name of the user-written generation application load module responsible for processing the virtual resources defined by this NCPNAU definition statement.

Code on: NCPNAU (page 2-330).

VIRTUAL

```
      ┌─ VIRTUAL=NO ─────┐
►►─┤                  ├──────────────────►◄
      └─ VIRTUAL=─┬─ YES ─┬─┘
                  └─ NO ──┘
```

For user-written programmed resources and Internet Protocol. Specifies whether NCP is to support programmed resources in the line group.

Code on: GROUP (page 2-139).

VOSDEACT

```
      ┌─ VOSDEACT=NO ─────┐
►►─┤                   ├─────────────────►◄
      └─ VOSDEACT=─┬─ YES ─┬─┘
                   └─ NO ──┘
```

Specifies whether VR deactivation will be initiated when a virtual route (VR) goes out-of-sequence.
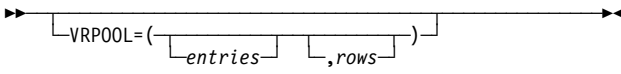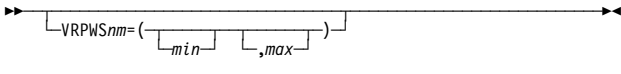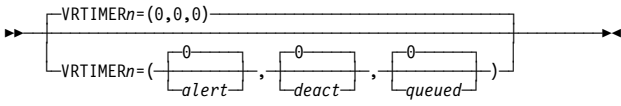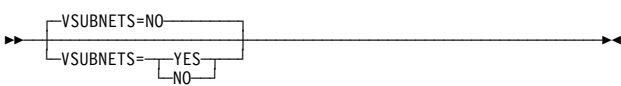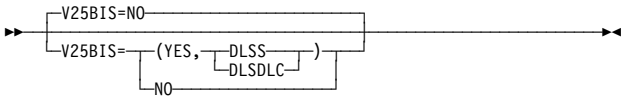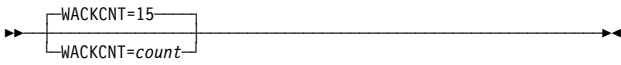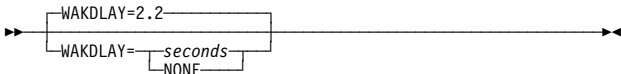
Code on: BUILD (page 2-61).

VPACING

```
      ┌─ VPACING=(0,,ADAP) ────────────────┐
►►─┤                                    ├─►◄
      │            ┌─ 0 ──────┐   ┌─ ADAP ──┐│
      └─ VPACING=(─┼──────────┼─,─┼─,─┼─ ADAP ──┤)┘
                   └─ number ─┘   └vtam┘└─ FIXED ─┘
```
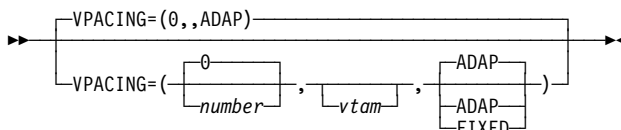
Overrides the batch mode window size or the adaptive pacing algorithm NCP uses to determine window size for this logical unit.

Code on: GROUP, LINE, PU, LU (page 2-309).

Keyword	Description

VPRINT

```
      ┌─VPRINT=NO──────────────┐
▶▶────┤                        ├────◀◀
      └─VPRINT=─┬─YES─┬─────────┘
                └─NO──┘
```

Specifies whether this device is to be treated as a virtual printer.

Code on: GROUP, LINE, CLUSTER, TERMINAL (page 2-472).

VR*n*

```
▶▶──────┬────────────────┬────────────◀◀
        └─VR n=ernumber ──┘
```

Specifies the virtual route to explicit route mapping that NCP uses when activating virtual routes.

Code on: PATH (page 2-357).

VRPOOL

```
▶▶────┬──────────────────────────────────┬────◀◀
      └─VRPOOL=(─┬─────────┬──┬───────┬─)─┘
                 └─entries─┘  └─,rows─┘
```

Defines the number of entries in the virtual route pool for NCP and specifies how many extra rows the flow control parameter table is to contain.

Code on: BUILD (page 2-61).

VRPWS*nm*

```
▶▶────┬──────────────────────────────┬────◀◀
      └─VRPWS nm=(─┬─────┬─┬──────┬─)─┘
                   └─min─┘ └─,max─┘
```

Specifies the minimum and maximum virtual route pacing window sizes.

Code on: PATH (page 2-358).

VRTIMER*n*

```
      ┌─VRTIMER n=(0,0,0)──────────────────────────────┐
▶▶────┤                                                ├────◀◀
      │          ┌─0───┐   ┌─0───┐    ┌─0────┐          │
      └─VRTIMER n=(─┴─────┴─,─┴─────┴─,─┴──────┴─)──────┘
                    alert     deact     queued
```

Controls the generation of alerts for congested virtual routes and the deactivation of held virtual routes. VRTIMER0, VRTIMER1, and VRTIMER2 define timer values for virtual routes with transmission priority 0, 1, and 2, respectively.

Code on: BUILD (page 2-62), NETWORK (page 2-338).

VSUBNETS

```
      ┌─VSUBNETS=NO──────────────┐
▶▶────┤                          ├────◀◀
      └─VSUBNETS=─┬─YES─┬─────────┘
                  └─NO──┘
```

Specifies whether variable subnetting is supported via NDF and dynamic reconfiguration.

Code on: BUILD (page 2-64).

V25BIS

```
      ┌─V25BIS=NO──────────────────────────┐
▶▶────┤                                    ├────◀◀
      └─V25BIS=─┬─(YES,─┬─DLSS───┬─)─┬──────┘
               │        └─DLSDLC─┘   │
               └─NO──────────────────┘
```

For V.25 bis. Specifies whether the modem associated with this line supports V.25 bis, and identifies which line protocol is used for the dial connection.

Code on: GROUP, LINE (page 2-294).

WACKCNT

```
      ┌─WACKCNT=15──────┐
▶▶────┤                 ├────◀◀
      └─WACKCNT=count ──┘
```

For BSC lines in network control mode. Specifies how many times NCP is to accept the wait-before-transmit positive acknowledgment from a BSC station before aborting the operation.

Code on: GROUP (page 2-139).

WAKDLAY

```
      ┌─WAKDLAY=2.2──────────┐
▶▶────┤                      ├────◀◀
      └─WAKDLAY=─┬─seconds─┬──┘
                 └─NONE────┘
```

For BSC lines in network control mode. Specifies how long NCP can delay before responding to message text received from any device in the line group.

Code on: GROUP (page 2-140).

Keyword	Description

WTTYEOB

```
►►──┬──────────────────┬──►◄
    └─WTTYEOB=chars─┘
```

For WTTY lines in network control mode. Specifies, in hexadecimal, the EOB sequence for WTTY teleprinters connected to lines in this group.

Code on: GROUP (page 2-140).

WTTYEOT

```
►►──┬──────────────────┬──►◄
    └─WTTYEOT=chars─┘
```

For WTTY lines in network control mode. Specifies, in hexadecimal, the EOT sequence for WTTY teleprinters connected to lines in this group.

Code on: GROUP (page 2-140).

XBREAK

```
       ┌─XBREAK=3─────────────┐
►►──┼──────────────────────┼──►◄
    └─XBREAK=─┬─3─────────┬─┘
             ├─char_times─┤
             └─NONE───────┘
```
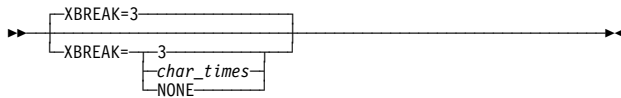
For start-stop lines in network control mode. Specifies the length, in character times, of the break signal NCP is to send to interrupt transmission from a terminal.

Code on: BUILD (page 2-64).

XID

```
►►──┬──────────────┬──►◄
    └─XID=─┬─YES─┬─┘
          └─NO──┘
```

Specifies whether this physical unit is to receive and respond to an exchange identification (XID) while in normal disconnected mode.

Code on: GROUP, LINE, PU (page 2-426).

XIO

```
►►──┬──────────────────────────────────────┬──►◄
    └─XIO=(line,setmode,immed,─┬──────┬─)─┘
                              └─link─┘
```
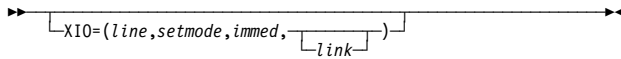
For user-written line control, NTRI, and frame relay. Specifies the entry points of execute-I/O service routines.

Code on: GROUP (page 2-141).

XMITDLY

```
►►──┬─────────────────────────┬──►◄
    └─XMITDLY─┬─seconds─┬──────┘
             └─NONE────┘
```

For SDLC lines in network control mode. Specifies, in seconds, how long NCP is to delay its initial transmission after answering an incoming call.

Code on: GROUP (page 2-142).

XMONLNK

```
►►──┬──────────────────┬──►◄
    └─XMONLNK=─┬─YES─┬─┘
             └─NO──┘
```

For SDLC peripheral links. Specifies whether the SSCP monitor mode function (SMMF) is to initially activate the link station.

Code on: GROUP, LINE, PU (page 2-427).

XTWXID

```
       ┌─XTWXID=YES─────────┐
►►──┼────────────────────┼──►◄
    └─XTWXID=─┬─YES─┬────┘
             └─NO──┘
```
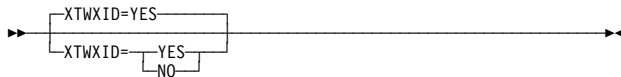
For TWX switched lines. Specifies whether, at line activation time, NCP is to transmit the identifier defined by TWXID on the BUILD definition statement.
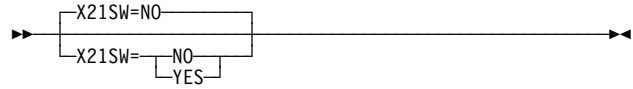
Code on: GROUP, LINE, TERMINAL (page 2-473).

X21NTWK

```
       ┌─X21NTWK=NO─────────┐
►►──┼────────────────────┼──►◄
    └─X21NTWK=─┬─NO───┬──┘
             ├─OLD──┤
             └─1984─┘
```

Specifies whether this group is attached to an X.21 network and, if so, the level of the X.21 network.

Code on: GROUP (page 2-142).

Keyword	Description

X21SW

```
       ┌─X21SW=NO───────────┐
►►─────┤                    ├─────────────────►◄
       └─X21SW=──┬─NO─┬─────┘
                 └─YES─┘
```

Specifies whether the lines in this group are X.21 switched lines. VTAM uses this information.

Code on: GROUP (page 2-143).

YIELD

```
       ┌─YIELD=YES──────────┐
►►─────┤                    ├─────────────────►◄
       └─YIELD=──┬─YES─┬─────┘
                 └─NO──┘
```

For BSC lines in network control mode. Defines the communication controller as the secondary device on a nonswitched point-to-point (contention) line.

Code on: GROUP, LINE (page 2-295).

List of Abbreviations, Glossary, Bibliography, and Index

List of Abbreviations

ACR	abandon-call-and-retry		**DLCI**	data link connection identifier
ACR	accelerated carriage return		**DMPX**	data multiplex
ACU	automatic calling unit		**DR**	dynamic reconfiguration
ALCI	airline line control integration		**DSAP**	destination service access point
ANA	assign network address		**DSE**	data switching equipment
ANR	automatic network routing		**DSR**	data set ready
ANS	automatic network shutdown		**DTE**	data terminating equipment
ANSI	American National Standards Institute		**EIB**	error information block
APPN	Advanced Peer-to-Peer Networking		**EOB**	end of block
ARP	Address Resolution Protocol		**EOT**	end of transmission
ASCII	American Standard Code for Information Interchange		**EP**	Emulation Program
		|	**EPSA**	Emulation Program Standalone
BAN	boundary access node		**ESCP**	ESCON processor
BECN	backward explicit congestion notification		**ESS**	Ethernet-type subsystem
BSB	boundary session control block		**FCS**	frame check sequence
BSC	binary synchronous communication		**FH**	frame handler
CBP	controller bus processor		**FHSP**	frame handler subport
CBSP	controller bus and service processor		**FIFO**	first in first out
CBTRP	control bus and Token-Ring processor		**FRMR**	frame reject
CCLID	called-calling-ID		**FRTE**	frame-relay terminating equipment
CCU	central control unit		**FVT**	function vector tables
CDROM	cross-domain resource manager		**HPR**	high performance routing
CD-ROM	compact disk read-only memory		**HPTSS**	High Performance Transmission Subsystem
CDRSC	cross-domain resources		**HSCB**	half-session control block
CDS	configuration data set		**HSS**	high-speed scanner
CLA	communication line adapter		**ICA**	integrated communication adapter
CLIP	Calling Line Identification Presentation		**IDNX**	integrated digital network exchange
CLIR	Calling Line Identification Restriction		**IEEE**	Institute of Electrical & Electronic Engineers
CLP	communication line processor		**InARP**	Inverse Address Resolution Protocol
| **CIR**	committed information rate		**IP**	Internet Protocol
CP	control point		**IPL**	initial program load
CPS	call progress signal		**ISDN**	integrated services digital network
CR	command reject		**ITB**	intermediate text block
CRI	call request with identification		**ITU-TS**	International Telecommunication Union - Telecommunication Standardization Sector.
CRN	call request with number			
CSS	connectivity subsystem		**JCL**	job control language
| **DAA**	Data Access Arrangement		**KVT**	keyboard vector table
DCE	data circuit-terminating equipment		**LA**	line adapter
DDI	Direct Dialing Inward		**LAN**	local area network

X-3

Abbreviations

LCST	line control selection table		**RECMS**	record maintenance statistics
LIC	line interface coupler		**REX**	route extension
LKP	link IPL ports		**RIP**	Routing Information Protocol
LLC	logical link control		**RIP-1**	Routing Information Protocol Version 1
LMI	local management interface		**RIP-2**	Routing Information Protocol Version 2
LOBQ	link outbound queue		**RNAA**	request network address assignment
LPDA	link problem determination aid		**RNR**	receive not ready
LU	logical unit		**RPN**	remote party number
MAC	medium access control		**RPQ**	request for price quotation
MB	megabyte		**RR**	receive ready
MLTG	multilink transmission group		**RSA**	remote subaddress
MOSS	maintenance and operator subsystem		**RTP**	Rapid Transport Protocol
MSN	Multiple Subscriber Number		**RTS**	request to send
MTA	multiple terminal access		**RU**	request/response unit
MTU	maximum transmission unit		**SAP**	service access point
NAU	network addressable units		**SCF**	secondary control field
NCP	Network Control Program		**SDLC**	synchronous data link control
NCST	NCP connectionless SNA transport		**SIT**	scanner interface trace
NDF	NCP/EP definition facility		**SKVT**	statement and keyword vector table
NDM	normal disconnect mode		**SLU**	secondary logical unit
NMVT	network management vector table		**SMMF**	SSCP monitor mode function
NNT	network names table		**SNA**	Systems Network Architecture
NPA	network performance analyzer		**SNBU**	switched network backup
NPM	NetView Performance Monitor		**SNI**	SNA network interconnection
NPSI	X.25 NCP Packet Switching Interface		**SOT**	service order table
NRF	Network Routing Facility		**SPS**	service and power support
NRZI	non-return-to-zero (inverted)		**SSAP**	source service access point
NTA	Network Traffic Analyzer		**SSCP**	system services control point
NTO	Network Terminal Option		**SSP**	System Support Programs
NTRI	NCP/Token-Ring interconnection		**TCP/IP**	Transmission Control Protocol/Internet Protocol
NTT	Nippon Telegraph and Telephone			
NVT	network vector table		**TESP**	terminating equipment subport
ODLC	outboard data link control		**TG**	transmission group
PAD	packet assembler/disassembler		**TGB**	transmission group control block
PCID	procedure correlation identifier		**TGN**	transmission group number
PDF	parallel data field		**TIC**	Token-Ring interface coupler
PEP	partitioned emulation programming		**TRP**	Token-Ring processor
PIU	path information unit		**TRSS**	Token-Ring subsystem
PLU	primary logical unit		**TRT**	transit routing table
PU	physical unit		**TSCP**	telecommunication service console processor
PVC	permanent virtual circuit			
			TSS	transmission subsystem

TTD	temporary text delay	**VSE**	Virtual Storage Extended
TWX	teletypewriter exchange service	**VTAM**	Virtual Telecommunications Access Method
UACB	user adapter control block	**WTTY**	World Trade Teletypewriter
UDP	User Datagram Protocol	**XID**	exchange identification
VLB	virtual line block	**XRF**	extended recovery facility
VR	virtual route		

Glossary

This glossary includes terms and definitions from:

- The *IBM Dictionary of Computing* (New York; McGraw–Hill, Inc., 1994)

- 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 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.

A

ACB. Adapter control block.

ACF/TAP. Advanced Communications Function/Trace Analysis Program. Synonymous with *TAP*.

ACF/TCAM. Advanced Communications Function for the Telecommunications Access Method. Synonym for *TCAM*.

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.

ACTLINK. Activate link.

ACTPU. Activate physical unit. In SNA, a command used to start a session on a physical unit.

ACU. Automatic calling 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* and *virtual route 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*. See *pacing*, *session-level pacing*, and *virtual route pacing*.

adaptive session pacing. Synonym for *adaptive session-level pacing*.

address mask. For internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an internet address. Synonymous with *subnet mask* and *subnetwork mask*.

address resolution. A method for mapping network-layer addresses to media-specific addresses. See also *Address Resolution Protocol (ARP)*.

Glossary

Address Resolution Protocol (ARP). A protocol that dynamically maps between Internet addresses, baseband adapter addresses, X.25 addresses, and Token-Ring adapter addresses on a local area network.

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.

adjacent NCPs. Network control programs (NCPs) that are connected by subarea links with no intervening NCPs.

adjacent nodes. Two nodes connected together by at least one path that connects no other node. (T)

adjacent subareas. Two subareas connected by one or more links with no intervening subareas. See also *subarea*.

Advanced Communications Function (ACF). A group of IBM licensed programs, principally VTAM, TCAM, NCP, and SSP, that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

Advanced Communications Function/Trace Analysis Program (ACF/TAP). An SSP program service aid that assists in analyzing trace data produced by VTAM, TCAM, and NCP and provides network data traffic and network error reports. Synonymous with *Trace Analysis Program (TAP)*.

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.

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 program, a high priority event that warrants immediate attention.

American National Standards Institute (ANSI). An organization consisting of producers, consumers, and general interest groups, that establishes the procedures by which accredited organizations create and maintain voluntary industry standards in the United States. (A)

ANA. Assign network address.

analog. (1) Pertaining to data consisting of continuously variable physical quantities. (A) (2) Contrast with *digital*.

AND operation. Synonym for *conjunction*.

ANSI. American National Standards Institute.

APAR. Authorized program analysis report.

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.

APPN. Advanced Peer-to-Peer Networking.

APPN node. See *Advanced Peer-to-Peer Networking (APPN) node*.

ARP. Address Resolution Protocol.

ASCII (American Standard Code for Information Interchange). The standard code, using a coded character set consisting of 7-bit coded characters (8-bit 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)

Note: IBM has defined an extension to ASCII code (characters 128–255).

asynchronous (ASYNC). (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.

auto-call. Synonym for *automatic calling*.

automatic calling. A feature that permits a station to initiate a connection with another station over a switched line without operator action. Synonymous with *auto-call*.

automatic calling unit (ACU). A dialing device that permits a computer to automatically dial calls over a network.

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.

BER. (1) Box event record. (2) Box error record.

BF. Boundary function.

BHSET. Block handler routines.

binary synchronous communication (BSC). 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. 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*.

blocking of PIUs. In SNA, an optional function of path control that combines multiple path information units (PIUs) in a single basic transmission unit (BTU).

Note: When blocking is not done, a BTU consists of one PIU.

BLU. Basic link unit.

boundary access node (BAN). A router (such as the IBM 6611) that provides its attached LAN-based SNA peripheral nodes direct frame-relay access to a subarea boundary node (such as an IBM 3745 or an IBM 3746 Model 900).

boundary function. (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 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.

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) See also *gateway*.

broadcast. (1) Transmission of the same data to all destinations. (T) (2) Simultaneous transmission of data to more than one destination.

broadcast address. In SDLC, a station address (eight 1's) reserved as an address common to all stations on a link.

BSC. Binary synchronous communication.

BTU. Basic transmission unit.

C

CA. (1) Channel adapter. (2) Channel attachment.

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)

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. An organization (one of four permanent organs of the International Telecommunication Union [ITU], headquartered in Geneva, Switzerland) that is concerned with the problems relating to international telephony and telegraphy. The CCITT Plenary Assembly meets at regular intervals to prepare a list of technical questions related to telephone and telegraph services. The Assembly assigns these questions to study groups, which then prepare recommendations to be presented at the next plenary meeting. Approved recommendations are published for the use of engineers, scientists, and manufacturers around the world.

CCU.   Central control unit.

CDRM.   Cross-domain resource manager.

CDS.   (1) Control data set. (2) Configuration data set.

channel.   (1) A path along which signals can be sent, for example, data channel, output channel. (A) (2) A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment. See *input/output channel*.

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. See also *link* and *subarea link*.

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.

CLP.   Communication line processor.

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*.

CMS.   Conversational monitor system.

communication adapter.   (1) A circuit card with associated software that enables a processor, controller, or other device to be connected to a network. (2) A mechanism that enables communication facilities to be attached to host processors.

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.

communication line processor (CLP).   In a communications controller, the processor that manages telecommunications lines.

configuration report program (CRP).   An SSP utility program that creates a configuration report listing network resources and resource attributes for networks with NCP, EP, PEP, or VTAM.

congestion.   See *network congestion*.

connection network.   A representation within an APPN network of a shared-access transport facility (SATF), such as a Token Ring, that allows nodes identifying their connectivity to the SATF by a common virtual routing node to communicate without having individually defined connections to one another.

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.

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 NAU 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 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 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.

conversational monitor system (CMS). A virtual machine operating system that provides general interactive time sharing, problem solving, and program development capabilities, and operates only under control of the VM/370 control program.

coupler. A device that connects a modem to a telephone network.

cross-domain. In SNA, pertaining to control or resources involving more than one domain.

cross-domain resource (CDRSC). In VTAM, synonym for *other-domain resource*.

cross-domain resource manager (CDRM). In VTAM, the function in the system services control point (SSCP) that controls initiation and termination of cross-domain sessions.

cross-network. In SNA, pertaining to control or resources involving more than one network.

cross-network session. An LU-LU or SSCP-SSCP session whose path traverses more than one SNA network.

CRP. Configuration report program.

CSS. Connectivity subsystem.

CWALL. An NCP threshold of buffer availability, below which the NCP will accept only high-priority path information units (PIUs).

D

data check. (1) An operation used to verify data quality or data integrity. (T) (2) A synchronous or asynchronous indication of a condition caused by invalid data or incorrect positioning of data. Some data checks can be suppressed.

data link connection identifier (DLCI). The numeric identifier of a frame-relay subport or PVC segment in a frame-relay network. Each subport in a single frame-relay port has a unique DLCI. The following table, excerpted from the American National Standards Institute (ANSI) Standard T1.618 and the International Telegraph and Telephone Consultative Committee (CCITT) Standard Q.922, indicates the functions associated with certain DLCI values:

DLCI Values	Function
0	in-channel signaling
1–15	reserved
16–991	assigned using frame-relay connection procedures
992–1007	layer 2 management of frame-relay bearer service
1008–1022	reserved
1023	in-channel layer management

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 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 *higher level*, *packet level*, and *physical level*. (2) In X.25 communications, synonym for *link level*.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

Glossary

data terminal equipment (DTE). That part of a data station that serves as a data source, data sink, or both. (I) (A)

datagram. (1) In packet switching, a self-contained packet, independent of other packets, that carries information sufficient for routing from the originating data terminal equipment (DTE) to the destination DTE without relying on earlier exchanges between the DTEs and the network. (I) (2) In TCP/IP, the basic unit of information passed across the Internet environment. A datagram contains a source and destination address along with the data. An Internet Protocol (IP) datagram consists of an IP header followed by the transport layer data. See also *packet* and *segment*.

DCE. Data circuit-terminating equipment.

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*.

definition statement. A type of instruction that defines a resource to the NCP. See Figure X-1 and Figure X-2.

```
                         operands
         ┌──────────────────┴──────────────────┐
      suboperands                        suboperands
       ┌───┴───┐                          ┌───┴───┐
 ┌────────┐ ┌──────┐                       ┌─────┐
 │ START  │ │A,(B,C),│  KEYWORD1=D,  KEYWORD2=(E,F)
 └────────┘ └──────┘
   │            │                             │
 statement   positional                    keyword
 identifier  operands                      operands
 └────────────────────────┬────────────────────────┘
                      statement
```
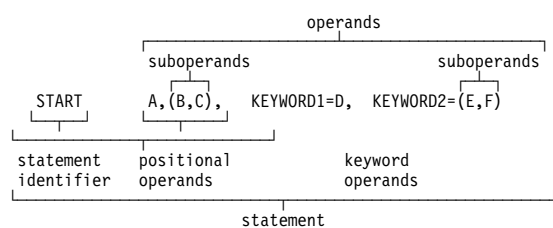
Figure X-1. Example of a Language Statement

```
definition    keyword operand
statement    ┌──────┴──────┐
identifier         suboperands
 ┌────┐         ┌──────┐
 │LINE│   AUTO=(YES,32)
 └────┘         └──────┘
   │
   └──── definition statement ────┘
```
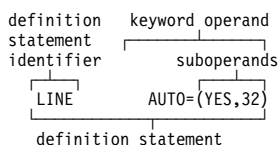
Figure X-2. Example of an NCP Definition Statement

destination address. A code that identifies the location to which information is to be sent.

destination service access point (DSAP). In SNA and TCP/IP, a logical address that allows a system to route data from a remote device to the appropriate communications subport. Contrast with *source service access point (SSAP)*.

device control character. A control character used to specify a control function for peripheral devices associated with a system. (T)

dial-out. Pertaining to the direction in which a switched connection is requested by a host or an NCP.

DLCI. Data link connection identifier.

DSAP. Destination service access point.

DSR. Data set ready.

DTE. Data terminal equipment. (A)

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 path update. The process of changing the network path for sending information without regenerating complete configuration tables.

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 consisting of 8-bit coded characters. (A)

ECB. Event control block.

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 *subarea address*.

emulation mode. The function of a network control program that enables it to perform activities equivalent to those performed by a transmission control unit. Contrast with *network control mode*.

Emulation Program (EP). An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. See also *network control program*.

end-of-transmission (EOT) character. (1) A transmission control character used to indicate conclusion of a transmission that may have included one or more texts and associated message headings. (I) (A) (2) In binary synchronous communication, the transmission control character usually used to end communication.

Enterprise Systems Connection (ESCON). A set of IBM products and services that provide a dynamically connected environment within an enterprise.

EOT. The end-of-transmission character. (A)

EP. (1) Emulation Program. (2) Entry point.

ER. (1) Explicit route. (2) Exception response.

ESCON. Enterprise Systems Connection.

ESCON processor (ESCP). A processor within the 3746 Model 900 connectivity subsystem (CSS) that performs ESCON function.

ESCP. ESCON processor.

Ethernet. A 10-Mbps baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

Ethernet-type LAN. A local area network that uses either the Ethernet Version 2 or IEEE 802.3 protocol.

event control block (ECB). A control block used to represent the status of an event.

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.

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)*.

extended binary-coded decimal interchange code (EBCDIC). A coded character set of 256 8-bit characters.

extended network addressing. The network addressing system that splits the address into an 8-bit subarea and a 15-bit element portion. The subarea portion of the address is used to address host processors or communication controllers. The element portion is used to permit processors or controllers to address resources.

extended recovery facility (XRF). A facility that minimizes the effect of failures in MVS, VTAM, the host processor, or high availability applications during sessions between high availability applications and designated terminals. This facility provides an alternate subsystem to take over sessions from the failing subsystem.

F

fallback. In an IBM 3745 Communication Controller with twin central control units (CCUs) in standby or backup mode, the process by which buses are switched from the failing CCU to the active CCU (backup mode) or the idle CCU (standby mode) to recover the path of communication flow in the failed CCU. See also *switchback.*.

fanout. A feature that allows several data terminal equipment (DTEs) to share the same modem. Only one DTE can transmit at a time.

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.

FHSP. Frame handler subport.

FIFO. First-in-first-out. (A)

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 *adaptive session-level pacing*, *pacing*, and *session-level 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 handler (FH). Synonym for *frame-relay frame handler (FRFH)*.

frame handler subport (FHSP). The access point of a frame-relay frame handler to a PVC segment. Frame handler subports function in pairs; frames enter the frame handler through one frame handler subport and exit through the other. Contrast with *terminating equipment subport*. See also *subport set*.

frame level. See *link level*.

frame relay. (1) An interface standard describing the boundary between a user's equipment and a fast-packet network. In frame-relay systems, flawed frames are discarded; recovery comes end-to-end rather than hop-by-hop. (2) A technique derived from the integrated services digital network (ISDN) D-channel standard. It assumes that connections are reliable and dispenses with the overhead of error detection and control within the network.

frame-relay frame. The frame-relay frame structure defined by American National Standards Institute (ANSI) Standard T1.618.

frame-relay frame handler (FRFH). (1) The function in a frame-relay node that routes (or switches) frames along a permanent virtual circuit (PVC). A frame handler receives frames from an adjacent frame-relay node and uses the DLCI to forward them to the next node on the PVC. Synonymous with *frame handler*. See also *frame-relay switching equipment support* and *frame-relay terminating equipment*. (2) In NCP, the function that switches frames between frame handler subports on an internal PVC segment. The NCP frame handler function can also switch frames to the frame-relay terminating equipment function.

frame-relay network. A network that consists of frame-relay frame handlers (FRFH) and in which frames are passed from one frame-relay terminal equipment (FRTE) station to another through a series of one or more FRFHs.

frame-relay physical line. The physical connection between two frame-relay nodes. A frame-relay physical line can simultaneously support PVC segments for both the frame-handler and terminating-equipment functions. In NCP, a frame-relay physical line is defined as a non-switched duplex line.

frame-relay switching equipment (FRSE). See *frame-relay switching equipment (FRSE) support*.

frame-relay switching equipment (FRSE) subport set. The set of primary and, optionally, substitute frame handler subports (FHSP) that comprise those used for a given frame-relay segment set.

frame-relay switching equipment (FRSE) support. In NCP, a set of frame-relay functions that include the frame-relay frame handler function and the local management interface (LMI) function. These functions are defined by American National Standards Institute (ANSI) Standards T1.617 and T1.618 and International Telegraph and Telephone Consultative Committee (CCITT) Standards Q.922 and Q.933. NCP provides additional functions, including performance measurement and enhanced reliability, that are not defined by ANSI or CCITT standards.

frame-relay terminal equipment. A device that can connect to a frame-relay network and provide the frame-relay terminating equipment function. See also *frame-relay frame handler* and *frame-relay terminating equipment*.

frame-relay terminating equipment (FRTE). The function at the end of a frame-relay permanent virtual circuit (PVC). Frame-relay terminating equipment provides higher-layer protocols with access to a frame-relay network through terminating equipment subports (TESPs). It does this by (a) adding frame-relay frame headers to data for another protocol and sending the frames to adjacent frame-relay nodes, and (b) receiving frames from adjacent frame-relay nodes and removing the frame headers. See also *frame-relay frame handler*, *frame-relay switching equipment support*, and *frame-relay terminal equipment*.

FRFH. Frame-relay frame handler.

FRSE. Frame-relay switching equipment.

full duplex (FDX). Synonym for *duplex*.

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 TCP/IP, a device used to connect two systems that use either the same or different communications protocols. (3) 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.

(4) 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.

gateway NCP. An NCP that performs address translation to allow cross-network session traffic. The gateway NCP connects two or more independent SNA networks. Synonymous with *gateway node*.

gateway node. Synonym for *gateway NCP*.

gateway SSCP. Synonym for *gateway VTAM*.

gateway VTAM. An SSCP that is capable of cross-network session initiation, termination, takedown, and session outage notification. A gateway VTAM is in session with the gateway NCP; it provides network name translation and assists the gateway NCP in setting up alias network addresses for cross-network sessions. Synonymous with *gateway SSCP*.

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 alert. A product-independent method of encoding alert data by means of both (a) code points indexing short units of stored text and (b) textual data.

group polling. A process whereby a single poll is sent to a collection point for a group of stations, inviting a response from any station in the group that has data to send.

H

half-duplex (HD, HDX). In data communication, pertaining to transmission in only one direction at a time. Contrast with *duplex*. See also *half-duplex operation* and *half-duplex transmission*.

half-session. A session-layer component consisting of the combination of data flow control and transmission control components comprising one end of a session. See also *session connector*.

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.

Hello. A protocol used by a group of cooperating, trusting routers to allow them to discover minimal delay routes.

hello datagram. (1) In the Internet suite of protocols, a hello message sent periodically to establish and test reachability between routers or between routers and hosts. The message is defined by the Hello protocol as an Interior Gateway Protocol (IGP). (2) In NCP IP support, an Internet Protocol (IP) datagram that establishes and verifies communication between an NCP IP router and its IP owner.

high-performance transmission subsystem (HPTSS). A high-speed line adapter that attaches to the IBM 3745 Communication Controller.

hop. To the routing layer, the logical distance between two nodes in a network.

host. 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 Internet 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 program interface (API) and a common application interface. See *boundary node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

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.

HPTSS. High-performance transmission subsystem.

hypertext link. A pointer from a location in an online book to another location in the same book or another book. When selected, a hypertext link enables you to move quickly to the new location containing related information. BookManager associates terms with related information such as the glossary, a message or code, an index entry, or a language element reference. Cross-references indicated by markup are automatically linked to the referenced location.

Glossary

I

I/O.   Input/output.

IBM Disk Operating System (DOS).   A disk operating system based on MS-DOS that operates with all IBM personal computers.

ICA.   Integrated communication adapter.

ID.   (1) Identifier. (2) Identification.

idle character.   A character transmitted on a telecommunication line that is not intended to represent data and does not result in an output operation at the accepting terminal.

IEEE.   Institute of Electrical and Electronics Engineers.

information (I) frame.   A frame in I format used for numbered information transfer.

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*.

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.

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.

internal PVC segment.   A permanent virtual circuit (PVC) segment between two frame handler subports in the same frame-relay node.

internet.   A collection of networks interconnected by a set of routers that allow them to function as a single, large network. See also *Internet*.

Standardization Sector.

International Telecommunication Union - Telecommunication.   The part of the International Telecommunication Union (ITU) that is responsible for developing recommendations for telecommunications.

Internet.   The internet administered by the Internet Architecture Board (IAB), consisting of large national backbone networks and many regional and campus networks all over the world. The Internet uses the Internet suite of protocols.

Internet address.   See *IP address*.

Internet Protocol (IP).   A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical network. However, this protocol does not provide error recovery and flow control and does not guarantee the reliability of the physical network.

Internet router.   A device that enables an Internet Protocol host to act as a gateway for routing data between separate networks that use a specific adapter.

IP.   Internet Protocol.

IP address.   The 32-bit address defined by the Internet Protocol, standard 5, Request for Comment (RFC) 791. It is usually represented in dotted decimal notation.

IP datagram.   In the Internet suite of protocols, the fundamental unit of information transmitted through an internet. It contains source and destination addresses, user data, and control information such as the length of the datagram, the header checksum, and flags indicating whether the datagram can be or has been fragmented.

IPL.   Initial program load.

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

katakana.   A character set of symbols used in one of the two common Japanese phonetic alphabets. Katakana is primarily used to write foreign words phonetically.

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

LAN. Local area network.

LAN segment. (1) Any portion of a LAN (for example, a bus or ring) that can operate independently, but that is connected to other parts of the network by means of bridges. (2) A ring or bus network without bridges.

LEN. Low-entry networking.

LIC. Line interface coupler.

limited resource. A connection facility that causes a session traversing it to be terminated if no session activity is detected for a specified period of time. See also *limited-resource session*.

limited-resource session. A session that traverses a limited-resource link. This session is terminated if no session activity is detected for a specified period of time.

line group. One or more telecommunication lines of the same type that can be activated and deactivated as a unit.

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*.

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)

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. 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). Synonymous with *data circuit*.

link integrity verification tests (LIVT). A set of operational procedures and messages that is defined by American National Standards Institute (ANSI) Standard T1.617 Annex D and International Telegraph and Telephone Consultative Committee (CCITT) Standard Q.933 Annex A and that is transferred over DLCI 0. This set of operational procedures and messages provides status and outage notification for frame-relay frame handler (FRFH) and frame-relay terminal equipment (FRTE) connections. See also *local management interface (LMI)*.

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 station. 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*.

LIVT. Link integrity verification tests.

LMI. Local management interface.

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 area network (LAN). (1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T) (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

See also *Ethernet* and *Token Ring*. (3) Contrast with *wide area network (WAN)*.

local management interface (LMI). See *local management interface (LMI) protocol*.

local management interface (LMI) protocol. (1) In NCP, a set of frame-relay network management procedures and messages used by adjacent frame-relay nodes to exchange line status information over DLCI X'00'. NCP supports both the American National Standards Institute (ANSI) and International Telegraph and Telephone Consultative Committee (CCITT) versions of LMI protocol. These standards refer to LMI protocol as *link integrity verification tests (LIVT)*. (2) Originally, a set of frame-relay network management procedures and messages used by frame-relay nodes to exchange line status information over DLCI X'03FF' (1023). This protocol is defined in *Frame-Relay Specification with Extensions*, a document based on proposed T1S1 standards, which are copyrighted by Digital Equipment Corporation, Northern Telecom, Inc., and StrataCom, Inc. This protocol is not compatible with the ANSI or CCITT version. In this context, the term *local management interface* is a synonym for *link integrity verification tests (LIVT)*.

locally administered address. In a local area network, an adapter address that the user can assign to override the universally administered address. Contrast with *universally administered address*.

logical link. 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.

logical unit (LU). A type of network accessible unit that enables end users to gain access to network resources and communicate with each other.

logical unit (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 program interface (API) consisting of structured verbs that are mapped into a product implementation.

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.

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.

LUCB. Logical unit control block.

M

MAC. Medium access control.

maintenance and operator subsystem (MOSS). A subsystem of an IBM communication controller, such as the 3725 or the 3720, that contains a processor and operates independently of the rest of the controller. It loads and supervises the controller, runs problem determination procedures, and assists in maintaining both hardware and software.

maintenance and operator subsystem extended (MOSS-E). A subsystem of the IBM 3745 Communication Controller that operates independently of the rest of the controller. It loads and supervises the controller, runs problem determination procedures, and assists in maintaining both hardware and software.

major node. In VTAM, a set of resources that can be activated and deactivated as a group. See *minor node*.

mask. (1) A pattern of characters used to control retention or elimination of portions of another pattern of characters. (I) (A) (2) To use a pattern of characters to control retention or elimination of portions of another pattern of characters. (I) (A)

maximum transfer unit (MTU). The maximum number of bytes that an Internet Protocol (IP) datagram can contain.

maximum transmission unit (MTU). In LANs, the largest possible unit of data that can be sent on a given physical medium in a single frame. For example, the MTU for Ethernet is 1500 bytes.

Mbps. One million bits per second.

medium access control (MAC). In LANs, the sub-layer 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) See also *logical link control protocol*.

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)

message switching. The process of receiving a message, storing it, and forwarding it to its destination unaltered. (T)

mixed-media multilink transmission group (MMMLTG). See *transmission group (TG)*.

MLTG. Multilink transmission group.

MMMLTG. Mixed-media multilink transmission group.

MOSS. Maintenance and operator subsystem.

MOSS-E. Maintenance and operator subsystem extended.

MTU. (1) Maximum transfer unit. (2) Maximum transmission unit.

multilink transmission group (MLTG). See *transmission group (TG)*.

Multiple Virtual Storage (MVS). See *MVS*.

MVS. Multiple Virtual Storage. Implies MVS/ESA and OS/390.

MVS/ESA product. Multiple Virtual Storage/Enterprise Systems Architecture.

N

native network. The subnetwork whose network identifier a node uses for its own network-qualified resource names.

NAU. (1) Network accessible unit. (2) Network addressable unit.

NCP. Network Control Program.

NCP connectionless SNA transport (NCST). An NCP function that allows a communication controller to transfer data across the SNA subarea routing network using TCP/IP protocols. The NCST function causes LU 0 sessions to be established between NCST logical units in the NCP and between an NCST logical unit in the NCP and SNA network link (SNALINK) logical units in the host processors.

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).

NCP/Token-Ring interconnection (NTRI). An NCP function that allows a communication controller to attach to the IBM Token-Ring Network and that provides both subarea and peripheral node data link control (DLC) services in the SNA network.

NCST. NCP connectionless SNA transport.

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*.

negotiable BIND. In SNA, a capability that allows two half-sessions to negotiate the parameters of a session when the session is being activated.

NETID. Network identifier.

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 accessible unit (NAU). A logical unit (LU), physical unit (PU), control point (CP), or system services control point (SSCP). It is the origin or the destination of information transmitted by the path control network. Synonymous with *network addressable unit*.

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 addressable unit (NAU). Synonym for *network accessible unit*.

network congestion. An undesirable overload condition caused by traffic in excess of what a network can handle.

network control mode. The mode in which a network control program can direct a communication controller to perform such activities as polling, device addressing, dialing, and answering. See also *emulation mode*.

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 identifier. (1) In TCP/IP, that part of the Internet 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.

network management vector transport (NMVT). A management services request/response unit (RU) that flows over an active session between physical unit management services and control point management services (SSCP-PU session).

network name. 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*.

network node (NN). Synonym for *Advanced Peer-to-Peer Networking (APPN) network node*.

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 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.

NMVT. Network management vector transport.

node. (1) In a network, a point at which one or more functional units connect channels or data circuits. (I) (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.

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 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. Contrast with *switched line*.

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.

NPM. NetView Performance Monitor.

NPSI. X.25 NCP Packet Switching Interface.

NRF. Network Routing Facility.

NRZ. Non-return-to-reference recording. (I) (A)

NRZ-1. Non-return-to-zero change-on-ones recording. (I) (A)

NRZI. Non-return-to-zero (inverted) recording. Synonym for *non-return-to-zero change-on-ones recording (NRZ-1)*.

NTO. Network Terminal Option.

NTRI. NCP/Token-Ring interconnection.

NTune. A set of programs (NTuneMON and NTuneNCP) that allow monitoring and tuning of active NCPs. See *NTuneMON* and *NTuneNCP*.

NTuneMON. A program that runs on NetView, and monitors NCPs that were activated, by VTAM, on the host where NTuneMON is running. See *NTune* and *NTuneNCP*.

NTuneNCP. A program that runs in the communications controller and, together with NTuneMON and VTAM provides interactive tuning capability of internal NCP resources. See *NTune* and *NTuneMON*.

O

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)

OS. Operating system.

other-domain resource. A representation for a logical unit that is owned by another domain and is referenced by a symbolic name, which can be qualified by a network identifier.

outboard data link control (ODLC). Data link control (DLC) processing performed by the 3746 Model 900 connectivity subsystem (CSS).

P

pacing. A technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See *session-level pacing*, *send pacing*, and *virtual route (VR) pacing*. See also *flow control*.

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. (1) 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) (2) In X.25, a data transmission information unit. A group of data and control characters, transferred as a unit, determined by the process of transmission. Commonly used data field lengths in packets are 128 or 256 bytes. (3) The field structure and format defined in the CCITT X.25 Recommendation.

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*, *higher 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 switching. 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)

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*.

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.

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)*.

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. See also *transmission header*.

PCID. Procedure-correlation identifier.

PDF. Parallel data field.

PEP. Partitioned emulation programming.

peripheral host node. A node that provides an application program interface (API) for running application programs but does not provide SSCP functions and is

not aware of the network configuration. The peripheral host node does not provide subarea node services. It has boundary function provided by its adjacent subarea. See *boundary node*, *host node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral link. In SNA, a link between a subarea and a peripheral node. See also *route extension (REX)*.

peripheral logical unit (LU). In SNA, a logical unit in a peripheral node.

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 (PU) in a peripheral node.

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 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 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. See also *logical unit (LU) services*.

physical unit type. In SNA, the classification of a physical unit (PU) according to the type of node in which it resides. The physical unit type is the same as its node type; that is, a type 1 physical unit resides in a type 1 node, and so forth.

PIU. Path information unit.

PLU. Primary logical unit.

point-to-point line. A switched or nonswitched tele-communication line that connects a single remote station to a computer. Contrast with *multipoint line*.

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. (I) (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 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.

port number. In Internet communications, the identification of an application entity to the transport service.

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*.

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*.

primary session. An extended recovery facility (XRF) session between the active application subsystem and a terminal user.

primary station. (1) In high-level data link control (HDLC), the part of a data station that supports the primary control functions of the data link, generates commands for transmission, and interprets received responses. (I) (2) In SNA, the station on an SDLC data link that is responsible for the control of the data link. There must be only one primary station on a data link. All traffic over the data link is between the primary

station and a secondary station. (3) Contrast with *secondary station*.

Note: Specific responsibilities assigned to the primary station include initialization of control signal interchange, organization of data flow, and actions to perform error control and error recovery functions.

procedure-correlation identifier (PCID). In SNA, a value used to correlate all requests and replies associated with a given procedure.

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*.

PU. Physical unit.

PVC. Permanent virtual circuit.

PVC segment. The connection between two frame-relay subports that may pass through a frame-relay network.

R

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 pacing. In SNA, the pacing of message units being received by a component. See also *send pacing*.

RECMS. Record maintenance statistics.

record maintenance statistics (RECMS). An SNA error event record built from an NCP or line error and sent unsolicited to the host.

redefinable line. A line that is in use and can be activated (defined using the USE keyword on the LINE definition statement). It can be changed to a spare line using NTuneMON with NTuneNCP.

remote. Pertaining to a system, program, or device that is accessed through a telecommunication line. Contrast with *local*. Synonym for *link-attached*.

request header (RH). The control information that precedes a request unit (RU). See also *request/response header (RH)*.

request unit (RU). A message unit that contains control information, end-user data, or both.

request/response header (RH). Control information associated with a particular RU. The RH precedes the request/response unit (RU) and specifies the type of RU (request unit or response unit).

request/response unit (RU). A generic term for a request unit or a response unit. See *request unit (RU)* and *response unit (RU)*.

response header (RH). A header, optionally followed by a response unit (RU), that indicates whether the response is positive or negative and that may contain a pacing response. See also *negative response*, *pacing response*, and *positive response*.

response unit (RU). A message unit that acknowl-edges 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 nega-tive, the response unit contains sense data defining the exception condition.

REX. Route extension.

RH. Request/response header.

RIP. Routing Information Protocol.

RNAA. Request network address assignment.

RNR. Receive not ready.

route. (1) An ordered sequence of nodes and trans-mission 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 spe-cific 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 refer-ence model network layer. Contrast with *bridge* and *gateway*. (3) In OSI terminology, a function that deter-mines a path by which an entity can be reached.

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 trans-mission header.

Routing Information Protocol (RIP). In the Internet suite of protocols, an interior gateway protocol used to exchange intradomain routing information and to deter-mine optimum routes between internet hosts. RIP determines optimum routes on the basis of route metrics, not link transmission speed.

routing table. A collection of routes used to direct datagram forwarding or to establish a connection. The information is passed among routers to identify network topology and destination feasibility.

RR. Receive ready.

RRT. Resource resolution table.

RU. Request/response unit.

S

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 char-acter recognition. (I) (A) (2) For the 3725 communi-cation controller, a processor dedicated to controlling a small number of telecommunication lines. It provides the connection between the line interface coupler hard-ware and the central control unit.

scanner interface trace (SIT). A record of the activity within the communication scanner processor (CSP) for a specified data link between an IBM 3725 Communi-cation Controller and a resource.

SCB. (1) Session control block. (2) String control byte.

SCF. Secondary control field.

SDLC. Synchronous Data Link Control.

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a par-ticular 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.

secondary station. A data station that executes data link control functions as instructed by the primary station. A secondary station interprets received commands and generates responses for transmission. Contrast with *primary station*.

send pacing. In SNA, pacing of message units that a component is sending. See also *receive pacing*.

service access point (SAP). (1) In Open Systems Interconnection (OSI) architecture, the point at which the services of a layer are provided by an entity of that layer to an entity of the next higher layer. (T) (2) A logical point made available by an adapter where information can be received and transmitted. A single service access point can have many links terminating in it.

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 *pacing* and *virtual route pacing*.

session limit. (1) The maximum number of concurrently active LU-LU sessions that a particular logical unit (LU) can support. (2) In NCP, the maximum number of concurrent line-scheduling sessions on a non-SDLC, multipoint line.

SIT. Scanner interface trace.

SLU. Secondary logical unit.

SMMF. SSCP monitor mode function.

SNA. Systems Network Architecture.

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. See *snapshot program*.

snapshot program. A trace program that produces output data only for selected instructions or for selected conditions. (I) (A)

SNBU. Switched network backup.

SNI. SNA network interconnection.

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*.

spare line. A line that is not in use and cannot be activated (defined using the USE keyword on the LINE definition statement). It can be changed to a redefinable line using NTuneMON with NTuneNCP, and then activated.

SS. (1) Start-stop. (2) Session services.

source service access point (SSAP). In SNA and TCP/IP, a logical address that allows a system to send data to a remote device from the appropriate communications subport. Contrast with *destination service access point (DSAP)*.

SSCP. System services control point.

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.

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. See also *binary synchronous transmission* and *synchronous data link control*.

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.

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 host node. A host node that provides both subarea function and an application program interface (API) for running application programs. It provides system services control point (SSCP) functions and subarea node services, and it is aware of the network configuration. See *boundary node*, *communication management configuration host node*, *data host node*, *host node*, *node*, *peripheral node*, and *subarea node*. See also *boundary function* and *node type*.

subarea link. In SNA, a link that connects two subarea nodes. See *channel link* and *link*.

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.

subnet. (1) In TCP/IP, a part of a network that is identified by a portion of the Internet address. (2) Synonym for *subnetwork*.

subnet mask. Synonym for *address mask*.

subnetwork. (1) Any group of nodes that have a set of common characteristics, such as the same network ID. (2) In the AIX operating system, one of a group of multiple logical network divisions of another network, such as can be created by the Transmission Control Protocol/Internet Protocol (TCP/IP) interface program. (3) Synonymous with *subnet*.

subnetwork mask. Synonym for *address mask*.

suboperand. One of multiple elements in a list comprising an operand. See also *definition statement*.

subport. (1) An access point for data entry or exit over a logical connection. The relationship between the physical line and the port is analogous to the relationship between the logical connection and the subport. (2) In a frame-relay network, the representation of a logical connection on a frame-relay physical line and the point where the logical connection attaches to the frame-relay frame handler (FRFH). Each subport on a physical line has a unique data link connection identifier (DLCI) and can represent an FRTE, FRFH, or LMI connection. See *frame handler subport (FHSP)* and *terminal equipment subport*.

subport set. In NCP, a set of frame handler subports linked by internal PVC segments. A subport set consists of two primary frame handler subports and an optional substitute frame handler subport for each primary.

switchback. In an IBM 3745 Communication Controller with twin central control units (CCUs) in backup mode, the process by which, after fallback, buses are moved back to the central control unit (CCU) that originally operated them.

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 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 network. Any network in which connections are established by closing switches, for example, by dialing.

switched network backup (SNBU). An optional facility that allows a user to specify, for certain types of physical units (PUs), a switched line to be used as an alternate path if the primary line becomes unavailable or unusable.

Synchronous Data Link Control (SDLC). 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) Contrast with *binary synchronous communication (BSC)*.

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 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

tailing. A feature on a multichannel modem that allows another modem link to be attached to one of the channels. See also *fanout*, *multitailed*, and *twin-tailed*.

TAP. Synonym for *ACF/TAP*.

TCAM. Telecommunications Access Method. Synonymous with *ACF/TCAM*.

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

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.

terminal component. A separately addressable part of a terminal that performs an input or output function, such as the display component of a keyboard-display device or a printer component of a keyboard-printer device.

terminal equipment subport. A subport that serves as a termination point on a virtual circuit.

terminating equipment subport (TESP). The endpoint of a frame-relay permanent virtual circuit (PVC); the point at which frame-relay terminating equipment has access to the PVC. A TESP provides higher-level functions with access to a frame-relay physical line. Each TESP in a single frame-relay port has a unique data link connection identifier (DLCI). Contrast with *frame handler subport*.

TESP. Terminating equipment subport.

TG. Transmission group.

TH. Transmission header.

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.

TIC. Token-Ring interface coupler.

Time Sharing Option (TSO). An operating system option; for the System/370 system, the option provides interactive time sharing from remote terminals.

timeout. (1) An event that occurs at the end of a pre-determined 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.

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. Each data station has an opportunity to acquire and use the token to control the medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) In LANs, a sequence of bits passed from one device to another along the transmission medium. When the token has data appended to it, it becomes a frame.

Token Ring. (1) According to IEEE 802.5, network technology that controls media access by passing a token (special packet or frame) between media-attached stations. (2) A FDDI or IEEE 802.5 network with a ring topology that passes tokens from one attaching ring station (node) to another. (3) See also *local area network (LAN)*.

Token-Ring interface coupler (TIC). An adapter that can connect a 3720, 3725, or 3745 Communication Controller to an IBM Token-Ring Network.

Token-Ring network. (1) A ring network that allows unidirectional data transmission between data stations, by a token passing procedure, such that the transmitted data return to the transmitting station. (T) (2) A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission.

Trace Analysis Program (TAP). Synonym for *Advanced Communications Function for the Trace Analysis Program (ACF/TAP)*.

Transmission Control Protocol (TCP). A communications protocol used in Internet and in any network that follows the U.S. Department of Defense standards for internetwork protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

transmission control unit (TCU). A communication control unit whose operations are controlled solely by programmed instructions from the computing system to which the unit is attached. No program is stored or executed in the unit. Examples are the IBM 2702 and 2703 Transmission Controls. Contrast with *communication controller*.

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 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 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 subsystem (TSS). A line adapter that attaches to the IBM 3745 Communication Controller.

TSO. Time Sharing Option.

TSS. Transmission subsystem.

twin-tailed. In NCP, pertaining to the attachment of a communication controller to two host processors. See also *fanout*, *multitailed*, and *tailing*.

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.

U

UDP.   User Datagram Protocol.

UNBIND.   In SNA, a request to deactivate a session between two logical units (LUs).   See also *session deactivation request*.   Contrast with *BIND*.

universally administered address.   In a local area network, the address permanently encoded in an adapter at the time of manufacture.   All universally administered addresses are unique.   Contrast with *locally administered address*.

User Datagram Protocol (UDP).   In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service.   It enables an application program on one machine or process to send a datagram to an application program on another machine or process.   UDP uses the Internet Protocol (IP) to deliver datagrams.

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-written generation application.   A user-written program that runs with the NCP/EP definition facility (NDF) during NCP generation.   It processes definition statements and operands.

V

V.25.   In data communications, a specification of the CCITT that defines the automatic answering equipment and parallel automatic calling equipment on the General Switched Telephone Network, including procedures for disabling of echo controlled devices for both manually and automatically established calls.

V.25 bis.   A procedure defined by CCITT that allows call establishment and data transfer to take place over the same link.   The support eliminates the need for two physical lines or ports when automatic call units (ACUs) are employed in a switched connection.

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).   (1) A virtual data processing system that appears to be at the exclusive disposal of a particular user, but whose functions are accomplished by sharing the resources of a real data processing system. (T)   (2) In VM/ESA, the virtual processors, virtual storage, virtual devices, and virtual channel subsystem allocated to a single user.   A virtual machine also includes any expanded storage dedicated to it.

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 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).   See also *explicit route (ER)*, *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 *pacing* and *session-level pacing*.

virtual route pacing response (VRPRS).   A nonsequenced, supervisory path information unit (PIU) that flows at network priority.   It may overtake VR-sequenced PIUs and consists of a transmission header with no basic information unit (BIU) data.

Virtual Storage Extended (VSE).   An IBM licensed program.   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.

VR.   Virtual route.

VSE.   Virtual Storage Extended.   The basic operating system support needed for a VSE-controlled installation.

VSE/ESA.   Virtual Storage Extended/Enterprise Systems Architecture.

VTAM.   Virtual Telecommunications Access Method.

W

WAN.   Wide area network.

wide area network (WAN).   (1) A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) (2) A data communications network designed to serve an area of hundreds or thousands of miles; for example, public and private packet-switching networks, and national telephone networks.   Contrast with *local area network (LAN)* and *metropolitan area network (MAN)*.

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) 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. (3) See *pacing window*.

window size.   The specified number of frames of information that can be sent before receiving an acknowledgment response.

X

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 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.

XID.   Exchange identification.

XMIT.   Transmit.

XRF.   Extended recovery facility.

Bibliography

NCP, SSP, and EP Library

The following paragraphs briefly describe the library for NCP, SSP, and EP. Other publications related to
| NTuneMON, VTAM, NPSI, NetView, NTO, and NRF are
| listed without the accompanying descriptions.

NCP and EP Reference (LY43-0029)

This book describes various aspects of the internal
| processing of NCP and EP (PEP or EP Standalone). It
provides information for customization and diagnosis.

NCP and EP Reference Summary and Data Areas
(LY43-0030)

This two-volume book provides quick access to often-
used diagnostic and debugging information about NCP
| and EP (PEP or EP Standalone).

NCP and SSP Customization Guide (LY43-0031)

This book helps users who are familiar with the internal
logic of NCP and SSP to modify these products. It
describes how to change NCP and SSP to support
stations that IBM-supplied programs do not support.

NCP and SSP Customization Reference (LY43-0032)

This book supplements the *NCP and SSP
Customization Guide*. It describes the resources and
macroinstructions provided by IBM for customizing NCP
and SSP.

NCP, SSP, and EP Diagnosis Guide (LY43-0033)

This book helps users isolate and define problems in
| NCP and EP (PEP or EP Standalone) using SSP. The
primary purpose of the book is to help the user interact
with the IBM Support Center to resolve a problem. In
addition, it explains some of the diagnostic aids and
service aids available with SSP.

NCP, SSP, and EP Trace Analysis Handbook
(LY43-0037)

This book describes how to use the trace analysis
program and how to read trace analysis program
output.

NCP, SSP, and EP Generation and Loading Guide
(SC31-6221)

This book provides detailed explanations of how to gen-
| erate and load NCP and EP (PEP or EP Standalone)

using SSP. It contains information for generating and
loading under MVS, VM, and VSE.

NCP, SSP, and EP Messages and Codes (SC31-6222)

This book is a reference book of abend codes issued by
| NCP and EP (PEP or EP Standalone), and messages
issued by the System Support Programs associated
with NCP. This information is also available through the
online message facility, an IBM OS/2 application avail-
able on diskette.

NCP, SSP, and EP Resource Definition Guide
(SC31-6223)

This book helps users understand how to define NCP
| and EP (PEP or EP Standalone), using SSP. It
describes functions and resources and lists the defi-
nition statements and keywords that define those func-
tions and resources.

NCP, SSP, and EP Resource Definition Reference
(SC31-6224)

This book helps users code definition statements and
| keywords to define NCP and EP (PEP or EP Stand-
| alone), using SSP. It also provides a quick reference of
definition statement coding order and keyword syntax.

NCP V7R7, SSP V4R7, and EP R14 Library Directory
(SC30-3971)

This book helps users locate information on a variety of
NCP, SSP, and EP tasks. It also provides a high-level
understanding of NCP, SSP, and EP and summarizes
| the changes to these products and to the library for
| NCP V7R7, SSP V4R7, and EP R14.

NCP V7R7 Migration Guide (SC30-3889)

This book helps users migrate an NCP generation defi-
nition from an earlier release to NCP V7R7. It also
describes how to add new functions for NCP V7R7.

*NCP Version 7 and X.25 NPSI Version 3 Planning and
Installation* (SC30-3470)

This book helps users plan and install support for X.25
lines in the 3745 or 3746 Model 900.

*NCP Version 7 and X.25 NPSI Version 3 Diagnosis,
Customization, and Tuning* (LY30-5610)

This book helps users diagnose, customize, and tune
X.25 lines in the 3745 or 3746 Model 900.

Other Networking Products' Libraries

The following publications provide cross-product information for NTuneMON, VTAM, NPSI, NetView, and NPM. For detailed information about these products, refer to the library for each.

Networking Library

Planning for NetView, NCP, and VTAM (SC31-8063)

Planning for Integrated Networks (SC31-8062)

ACF/NCP, ACF/SSP, EP, NTuneMON, and NPSI Softcopy Collection Kit (CD-ROM, LK2T-0414)

NTuneMON Library

NTuneMON V2R5 User's Guide (SC31-6266)

NTuneNCP Feature Reference (LY43-0039)

Related Publication

NCP Tuning with NTune, GG24-2520

VTAM Library

VTAM Migration Guide (GC31-6547)

VTAM Release Guide (GC31-6555)

Estimating Storage for VTAM (SK2T-6400)

VTAM Network Implementation Guide (SC31-6548)

VTAM Resource Definition Reference (SC31-6552)

VTAM Resource Definition Samples (SC31-6554, book and diskettes)

VTAM Customization (LY43-0068)

VTAM Operation (SC31-6549)

VTAM Operation Quick Reference (SX75-0207)

VTAM Messages and Codes (SC31-6546)

VTAM Licensed Program Specifications (GC31-6553)

VTAM Programming (SC31-6550)

VTAM Programming for LU 6.2 (SC31-6551)

VTAM Diagnosis (LY43-0069)

VTAM Data Areas for MVS/ESA (LY43-0071)

VTAM CMIP Services and Topology Agent Programming Guide (SC31-6544)

APPC Application Suite User's Guide (SC31-6532)

APPC Application Suite Administration (SC31-6533)

APPC Application Suite Programming (SC31-6534)

NPSI Library

X.25 NPSI Version 3 General Information (GC30-3469)

NCP Version 7 and X.25 NPSI Version 3 Planning and Installation (SC30-3470)

X.25 NPSI Version 3 Host Programming (SC30-3502)

NCP Version 7 and X.25 NPSI Version 3 Diagnosis, Customization, and Tuning (LY30-5610)

X.25 NPSI Version 3 Release 9 Data Areas (LY30-5627)

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 Planning Guide
(GC31-8226)

*TME 10 NetView for OS/390 Installation and Adminis-
tration 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 User's Guide (SC31-8241)

TME 10 NetView for OS/390 NGMF User's Guide
(SC31-8234)

*TME 10 NetView for OS/390 SNA Topology Manager
and APPN Accounting Manager Implementation Guide*
(SC31-8239)

TME 10 NetView for OS/390 Data Model Reference
(SC31-8232)

*TME 10 NetView for OS/390 APPN Topology and
Accounting Agent Guide* (SC31-8224)

| ## NTO Library

| *Network Terminal Option: R11 Planning, Migration, and
| Resource Definition Guide* (SC30-3347)

| *Network Terminal Option: R11 Diagnosis* (LY30-3194)

| ## NRF Library

| *NRF Planning* (SC27-0593)

| *NRF Migration, Resource Definition and Customization*
| (SC31-6203)

| *NRF Resource Definition and Customization*
| (SC30-3407)

| *NRF Diagnosis* (LY30-5597)

Related Publications

The following publications, though not directly related to
this book, may be helpful in understanding your
network.

*Internet Standard Subnetting Procedure (TCP/IP RFC
950)*

Network Design and Analysis General Information
(GC30-3495)

Remote Loading/Activation Guide (SA33-0161)

| *3745 and 3746 Service Processor Installation and Main-
| tenance* (SY33-2115)

| *TPF Concepts and Structures Manual* (GH20-7488)

IBM 3745 Communication Controller Publications

The following list shows selected publications for the
IBM 3745 Communication Controller.

IBM 3745 Communication Controller Introduction
(GA33-0138 for the 3745-130, 3745-150, and 3745-170)

IBM 3745 Communication Controller Introduction
(GA33-0092 for the 3745-210, 3745-310, 3745-410, and
3745-610)

| *IBM 3745 Communication Controller All Models and
| 3746 Model 900: Connection and Integration Guide*
| (SA33-0129)

| *3745 Communications Controller Models 130, 150, 160,
| and 170: Connection and Integration Guide*
| (SA33-0141)

*IBM 3745 Communication Controller (All Models): Prin-
ciples of Operation* (SA33-0102)

| *IBM 3745 Advanced Operations Guide* (SA33-0097)

| *3745 Communication Controller Models A, 3746 Nways
| Multiprotocol Controller Models 900 and 950: Planning
| Guide* (GA33-0457)

| *IBM 3745 Problem Determination Guide* (SA33-0096)

IBM 3746 Model 900 and Model 950 Publications

*IBM 3745 Communication Controller Models A, IBM
3746 Expansion Unit Model 900, IBM 3746 Nways
Multinetwork Controller Model 950 Overview*
(GA33-0180)

*IBM 3746 Nways Multinetwork Controller Model 950
User's Guide* (SA33-0356)

MVS/ESA Publications

MVS/ESA Device Validation Support (GC28-1617)

| *MVS/ESA Hardware Configuration Definition: User's*
| *Guide* (GC33-6457)

SNA Publications

The following publications contain information on SNA.

Systems Network Architecture Technical Overview
(GC30-3073)

Systems Network Architecture Management Services Reference (SC30-3346)

Systems Network Architecture Formats (GA27-3136)

TCAM Publications

| *TCAM Version 2 Base Installation Guide* (SC30-3132)

TCAM Version 2 Networking Installation Guide
(SC30-3153)

TCP/IP Publications

The following publications contain information on Transmission Control Protocol/Internet Protocol (TCP/IP).

General

The following list shows selected publications with general information on TCP/IP.

Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991 (SC31-6144)

Internetworking with TCP/IP Volume II: Implementation and Internals, Douglas E. Comer, Prentice Hall, Englewood Cliffs, New Jersey, 1991 (SC31-6145)

TCP/IP Introduction (GC31-6080)

IBM TCP/IP Tutorial and Technical Overview
(GG24-3376)

MVS Publications

The following list shows selected publications on TCP/IP for MVS.

IBM TCP/IP Version 2 Release 2.1 for MVS: Planning and Customization (SC31-6085)

IBM TCP/IP Version 2 Release 2.1 for MVS: User's Guide (SC31-6088)

VM Publications

The following list shows selected publications on TCP/IP for VM.

| *IBM TCP/IP Version 2 Release 4 for VM: Planning and Customization* (SC31-6082)

| *IBM TCP/IP Version 2 Release 4 for VM: User's Guide* (SC31-6081)

IBM OS/2 Publications

The following list shows selected publications on TCP/IP for IBM OS/2.

IBM TCP/IP Version 2.0 for OS/2: Installation and Administration (SC31-6075)

IBM TCP/IP Version 2.0 for OS/2: User's Guide (SC31-6076)

DOS Publications

The following list shows selected publications on TCP/IP for DOS.

| *IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration* (SC31-7047)

| *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*
| (SC31-7045)

Index

Index

C

Index

Index

Index

Q

Index

Tell Us What You Think!

Network Control Program
System Support Programs
Emulation Program
Resource Definition Reference

Publication No. SC31-6224-07

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 U.S.A., you can mail this
form postage free or fax it to us at 1-800-253-3520. Elsewhere, your local IBM branch office or represen-
tative will forward your comments or you may mail them directly to us.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

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.

Tell Us What You Think!
SC31-6224-07

IBM ®

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES
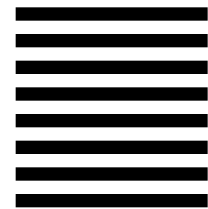
BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Design & Information Development
Dept. CGF/Bldg. 656
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

SC31-6224-07

IBM®